NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DESIGN AND IMPLEMENTATION OF A DECISION SUPPORT SYSTEM TO AID IN THE FORECASTING AND SCHEDULING OF ADMINISTRATIVE SCIENCES COURSES

by

Mark D. VanUs
and
William G. Reagle

March 1991

Thesis Advisor: Shu Sheng Liao
Co-Advisor: Daniel R. Dolk

Approved for public release; distribution is unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/ AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DCLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL *(If Applicable)* 37	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS *(city, state, and ZIP code)* Monterey, CA 93943-5000	7b. ADDRESS *(city, state, and ZIP code)* Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 6b. OFFICE SYMBOL *(If Applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(city, state, and ZIP code)*	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
Design and Implementation of a Decision Support System to Aid in the Forecasting and Scheduling of Administrative Sciences Courses

12. PERSONAL AUTHOR(S)
VanUs, Mark D., and Reagle, William G.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(year, month, day)*	15. PAGE COUNT
Master's Thesis	FROM TO	March 1991	123

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUBGROUP	Decision Support System; Database Management; Scheduling

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The Administrative Sciences Department's faculty and course scheduling process is complex and data intensive. The process is driven by student enrollment forecasts provided to the department by the Naval Postgraduate School's Program Administration Assistant. The department currently has no means for manipulating the forecasts.

This thesis develops a decision support system and its associated dBASE IV-based relational database for use by the Administrative Sciences Department in the forecasting and scheduling processes. The system consists of four major applications comprised of over 65 separate modules. The database management application allows for management and maintenance of the database files. The scheduling application generates teaching schedules. The report generating application produces all the reports required of the system by the department. Finally, the estimation application provides the decision support portion of the system by allowing "what-if" manipulation of the student enrollment data to see what impact there will be on teaching requirements.

The system was developed within a Systems Development Life Cycle framework. Due to the need to quickly produce a working copy of the system, individual modules were developed using a Version Development technique.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
[X] UNCLASSIFIED/UNLIMITED [ ] SAME AS RPT. [ ] DTIC USERS	Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
Prof. Shu Sheng Liao, Prof. Daniel R. Dolk	(408) 646-2505	AS/LC

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted SECURITY CLASSIFICATION OF THIS PAGE

All other editions are obsolete Unclassified

Approved for public release; distribution is unlimited.

Design and Implementation of a Decision Support System to Aid in the Forecasting and Scheduling of Administrative Sciences Courses

by

Mark D. VanUs
Captain, United States Army
B.B., Western Illinois University
and
William G. Reagle
Captain, United States Army
B.S., University of Pittsburgh

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS
from the
NAVAL POSTGRADUATE SCHOOL
March 1991

ABSTRACT

The Administrative Sciences Department's faculty and course scheduling process is complex and data intensive. The process is driven by student enrollment forecasts provided to the department by the Naval Postgraduate School's Program Administration Assistant. The department currently has no means for manipulating the forecasts.

This thesis develops a decision support system and its associated dBASE IV-based relational database for use by the Administrative Sciences Department in the forecasting and scheduling processes. The system consists of four major applications comprised of over 65 separate modules. The database management application allows for management and maintenance of the database files. The scheduling application generates teaching schedules. The report generating application produces all the reports required of the system by the department. Finally, the estimation application provides the decision support portion of the system by allowing "what-if" manipulation of the student enrollment data to see what impact there will be on teaching requirements.

The system was developed within a Systems Development Life Cycle framework. Due to the need to quickly produce a working copy of the system, individual modules were developed using a Version Development technique.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

The Naval Postgraduate School is a unique university where the student body is comprised wholly of U.S. and international military officers and DoD civilian employees. The mission of the Naval Postgraduate School is to conduct the advanced education of commissioned officers, and to provide such other instruction as may be prescribed to meet the needs of the naval service, and in support of research in order to sustain academic excellence. The school is authorized to confer Bachelor's, Master's, Engineer's and Doctor's degrees upon qualified graduates.

Members of the faculty are organized into eleven Academic Departments and three Interdisciplinary Academic Groups, each supervised by a chairman. Each department is responsible for the content and administration of its own unique academic programs. Faculty resource utilization is a major portion of the administrative workload within the departments. This thesis is concerned with the planning and management of faculty resource utilization within the Department of Administrative Sciences.

The Department of Administrative Sciences has primary responsibility for three academic programs, and awards three graduate degrees. The largest program consists of a group of

curricula which include Acquisition and Contract Management, Financial Management, Manpower/Personnel/Training Analysis, Material Logistics Support, Systems Inventory Management and Transportation Management. Other curricula associated with the Administrative Sciences Department are Computer Systems Management and Telecommunications Systems Management which will be combined into a new curriculum called Information Technology Management, beginning October 1991.

In order to fulfill the academic requirements of each of the curricula, the Administrative Sciences Department must schedule and coordinate, based on the number of students forecast to be enrolled in each curriculum, 84 courses of instruction and 67 department instructors. The department produces a standard matrix of required courses, a list of approved electives and the design of alternative emphasis areas for each of the curricula. The scheduling process is complicated by inaccurate student population forecasts, constantly changing student enrollment, and by students modifying their standard course matrix through course validation and through the process of choosing courses for their emphasis areas.

B. STATEMENT OF THE PROBLEM

Since the Naval Postgraduate School does not have direct control over the admission of students, the Administrative Sciences Department does not know with any accuracy what their student enrollment figures are going to be for any given

quarter in the planning cycle. The projected number of students in each class, as well as the curricula the students are enrolled in, frequently undergo substantial changes after initial teaching schedules have been produced. Due to these inaccuracies, current scheduling techniques utilized by the Administrative Sciences Department cannot adequately forecast enrollment, causing significant difficulty in scheduling courses and instructors. The 1989 summer planning cycle illustrates the effects of this problem. For several years preceding the summer quarter of 1989, the Administrative Sciences Department's "normal" input was about 110 students, plus or minus ten percent. Suddenly, with less than a month's notice, the Navy decided to fill all Naval Postgraduate School quotas to 100% and the department received an input of 135 students. This additional student load required teaching one extra section for every "core" course offered by the department and necessitated the hiring of additional faculty on very short notice.

One solution to this problem of predicting student input is to develop a system which can generate teaching schedules based on different scenarios of student inputs. This would allow the department to more accurately assess the impact of unforeseen enrollments on teaching requirements.

C. SCOPE

The purpose of this thesis is to develop a comprehensive and responsive microcomputer decision support system that will

support planners within the Administrative Sciences Department in forecasting and planning faculty resources to meet future instructional needs. It will develop a model that reflects the current scheduling environment and design a system to support the scheduling process. A working system will be delivered to the Administrative Sciences Department upon completion of system development.

D. METHODOLOGY

1. General

The existing system and environment in which relevant decisions are made were reviewed to determine the system requirements. Analysis of the current scheduling process showed that a major portion of the Educational Technician's time was used for data input and adjusting previous teaching schedules. It also identified a need for a local system which organization personnel at different levels can use to produce information at various degrees of aggregation. From this analysis and from the need to produce a working system quickly, a decision was made to utilize a Version Development technique for rapid prototyping within the systems development life cycle framework.

2. System Development Life Cycle

A system development life cycle consists of four phases to plan and control systems development projects. These phases are analysis, design, implementation and maintenance. (Whitten/Beatley/Barlow, 1989:p. 81)

During the analysis phase, information is gathered to define the scope of the project and determine its feasibility. The current system is then analyzed to identify problems and opportunities within the system, and user requirements are defined. The analysis phase of this project is described in Chapter II.

The design phase consists of both logical design and application design. In logical design a blueprint of the database needed to support all applications is produced. During application design the scope, control mechanisms, menu options and data presentation format for each application are developed. The design phase is discussed in Chapter III.

The implementation phase includes actual delivery, set up, training and the writing of manuals. This phase is briefly discussed in Chapter IV.

The final phase is to maintain and improve the system. Proper analysis and design can reduce the complexity and frequency of this phase which is addressed in Chapter V.

It is critical that the developed system be flexible, maintainable and meet the needs of the department. It must also be reliable and maintain data integrity. Failure to achieve objectives such as these is usually rooted in the analysis and design phases of the life cycle (Powers, Cheney, Crow, 1990:p. 55). This work emphasizes the analysis and design phases to ensure a successful implementation.

3. Version Development

Version development follows a simple principle: develop a complex system by breaking it into a series of parts, or modules, that can be separately designed and implemented. A basic module is implemented first and then other modules are added to provide additional functionality. Version development may not necessarily shorten the development effort but it does speed the process of installing a base version of the system. (Powers, Cheney, Crow, 1990: p. 793-794)

Upon completion of the analysis and design phases, modules were identified and then prioritized for implementation. The advantage of this approach is that the database is designed with the total system in mind. Each module, when implemented, supports the total system. To allow continuous user input during system development and design, and to ensure that user specifications were met, each module was designed and implemented with the full support and participation of the department's Associate Chair for Instruction. The modules were implemented in the following order: database management, schedule building, report generation and man-year estimation.

II. SYSTEM ANALYSIS

A. PRESENT SYSTEM

The present system used by the Administrative Sciences Department to schedule courses and instructors involves interaction between several organizations at the school. Those organizations are the Registrars Office, the Scheduler, other Curriculum Offices and other academic departments. Within the Administrative Sciences Department there is constant interaction between the Department Chairman, Associate Chair for Instruction, Educational Technician and the Curricular Offices for Administrative Sciences and Computer Technology.

The Naval Postgraduate School's Program Administration Assistant produces and disseminates student population forecasts. The specifics of how these forecasts are developed are beyond the scope of this thesis; however, they are the primary source of uncertainty with respect to course scheduling. It is for this reason that the Administrative Sciences Department wants to be able to perform "what-if" analysis with these forecasts.

The Program Administration Assistant's forecasts are combined with pre-enrollment data obtained from the curriculum offices to produce the "1st Iteration" report. The 1st Iteration is sent to all departments and is commonly received

10 to 11 weeks prior to the start of the quarter it refers to. The 1st Iteration is used by the Administrative Sciences Department to produce a preliminary course schedule. The department produces its initial Course Offerings (CO) report and Teaching Schedule by Discipline (TSD) report based on the preliminary course schedule.

During the time the department is producing these two reports, the Program Administration Assistant is updating the enrollment figures. The "2nd Iteration" report encapsulates those changes and is sent to all departments. The 2nd Iteration is commonly received seven to eight weeks prior to the start of the quarter it refers to.

Based on the changes between the 1st Iteration and the 2nd Iteration, which can be substantial, the department makes changes to their TSD report produced while waiting for the 2nd Iteration. The department then publishes their final Teaching Schedule. The entire system is driven by the Iteration reports and takes several weeks to complete. If the Iteration reports are inaccurate or late, the system is slowed even further.

B. USER REQUIREMENTS

The determination of user requirements, in terms of both data and functional requirements, is a critical step in systems analysis. User requirements for the Instruction Scheduling Program were obtained through interviews and discussions with all personnel who will interact with the

system. Additional requirements were identified by examining the current system's output and then working backwards to derive associated data and functional requirements.

1. Data requirements

After the data requirements were identified during the interviews and discussions, and indirectly from the output, they were then translated into data objects. Data objects are structured representations of the entities the user wants, or needs, to keep track of (Kroenke, Dolan, 1988:p. 88). The objects represent a conceptual view of the data that will eventually need to be stored in the database, that is, the database will contain instances of these objects. Appendix C contains the object diagram.

The **COURSE Object** contains all the pertinent data on the courses offered by the Administrative Sciences Department. Each course is uniquely identified by its course number. The data includes the desired class size, the number of lecture hours and lab hours, what discipline the course falls under and what quarter during each year the course is offered. There are over 100 courses offered by the department. The data was obtained from the NPS Course Catalog and the department's Educational Technician.

The **CLASS Object** is a specific instance of a course and contains information on the year and quarter in which a section of a course will be taught. It also contains information on the number of sections required per quarter and

the number of students enrolled in each section of a course. There are approximately 500 classes taught by the Administrative Sciences Department annually. The number of sections required is calculated using the projected student enrollment for that course and the maximum course size. The remaining data was obtained from the department's Educational Technician and the NPS Registrar Office.

The **DISCIPLINE Object** contains information on the nine different disciplines found within the Administrative Sciences Department. An example is the Information Systems discipline (367) which encompasses all of the courses that have IS as the first two letters of their course number. Each discipline is uniquely identified by its discipline number and contains courses relevant to the discipline area. This object is used in producing the teaching schedule by discipline report. The data were obtained from the department's Educational Technician.

The **COURSE CURRICULUM Object** contains information on the percentage of students from each curriculum that have historically taken a course during a certain quarter. This data is the heart of the scheduling and estimating portions of this system. There are approximately 250 records associated with this object. The raw student data were obtained from the Registrar's Office and then condensed to produce the percentages.

The **CURRICULUM Object** contains information on all ten of the curricula the Administrative Sciences Department determined to be critical to the system. Each curriculum is uniquely identified by its curriculum number. The data for each curriculum in this object include; the length in quarters, the number of students enrolled, the number and size of sections and the quarter(s) that each one starts its new sections. The data were obtained from the curricular offices.

The **SECTION Object** contains the data on each section within each curriculum. Each section is uniquely identified by its section name. Currently, there are 36 sections registered in the curricula. The data were obtained from the curricular offices.

The **FACULTY Object** contains data on all the faculty within the Administrative Sciences Department. Each faculty member is uniquely identified by their faculty ID. This object also contains the normal teaching load per quarter of each faculty member which is used in determining the teaching schedule. There are approximately 70 faculty members in the department. The data were obtained from the Administrative Sciences Department.

The **FACULTY AVAILABILITY Object** is used to maintain information on whether a faculty member will be available to teach classes during any given quarter of a given year. Research quarters, sabbaticals or any other event which would interfere with a faculty member's ability to teach would be

captured here. There are approximately 150 records to consider when determining faculty availability. The data were obtained from the Administrative Sciences Department.

The **FACULTY EXPERTISE Object** contains the courses each faculty member is qualified to teach. This information, combined with the faculty availability data, is used in determining who will instruct a class during a given quarter. The Administrative Sciences Department's faculty is qualified to teach approximately 250 courses (this number includes courses that can be taught by many faculty). The data were obtained from the Administrative Sciences Department.

The final object, **CLASS INSTRUCTOR**, contains data on the teaching schedule for each faculty member. Each record includes the course, the year, the quarter and the number of sections of the course a faculty member will teach. There are approximately 500 class instructor assignments annually. The data were obtained by relating the faculty data with the class data.

2. Functional Requirements

Functional requirements are equivalent to the application requirements of the system. The functional requirements for the Instruction Scheduling Program are portrayed as data flow diagrams which can be found in Appendix B.

a. Database Management Application

The functional requirements of the Database Management application consist of creating, modifying and deleting Curricula, Sections, Disciplines, Faculty and Courses (Appendix B, processes 1.1 through 1.5). Course Curriculum data are associated with the Curricula application, Class data with Courses, and Faculty Availability, Faculty Expertise and Class Instructor data with faculty. The data for this application will be furnished by the Associate Chair for Instruction within the Administrative Sciences Department, the department chairman, and the registrars office. The data, once collected, will remain within the department for use in this system only.

b. Schedule Application

The functional requirements of the schedule application consist of creating a template of courses to be offered during an academic year (Appendix B, process 1.10). In order to create this schedule the Educational Technician supplies the date which is then combined with the data stored in the database to create the schedule. The schedule is then reviewed and distributed to personnel within the department. From this schedule individual instructor notifications are also prepared. Required changes to the schedule can be made through data input to the database management application. The schedule application is then run again to produce the updated schedule.

c. Report Generating Application

The functional requirements for this application include displaying and printing seven separate reports (Appendix B, processes 1.6 through 1.9): Teaching Schedule By Discipline (TSD), Estimated Man-Year Report (EMY), Estimated Teaching Schedule by Discipline (ETSD), Quarterly Teaching Schedule (QTS), Course Offerings Report (CO), Teaching Schedule for One Discipline (TSOD) and Faculty Teaching Report (FTR). The formats of these reports are identical to those used in the current system.

The TSD report is generated by combining the date, which is provided by the Educational Technician, with data stored in the Faculty, Discipline, Course, Class and Class Instructor objects. This report is the primary tool for verifying and correcting faculty teaching assignments for the given year. The report is disseminated by the Educational Assistant to appropriate personnel in the department.

The EMY report is generated by combining the date with data stored in the Class and Class Instructor objects. This report reflects the latest man-year estimate run by the user. Modifications to this report must be made through the Estimation Application. This report is used for financial planning by the department.

The ETSD report is generated by combining the data stored in the Discipline, Course and Class objects with the student enrollment data utilized in the "what-if" man-year

estimation process. It lists the courses and the number of sections for each of those courses required to be taught based on the enrollment figure used. It is used by the Department Chairman and the Associate Chair for Instruction in conjunction with the EMY report for more detailed financial planning.

The QTS report is generated by combining the date with data stored in the Class object. This report is used by the Associate Chair for Instruction to assist in monitoring the teaching load during the specified quarter.

The CO report is generated by combining the date with data stored in the Class object. This report is distributed within the Administrative Sciences Department and to all other departments at the school. This report informs the departments of what courses the Administrative Sciences Department will offer in the specified quarter.

The TSOD report is generated in the same manner as the TSD report. This report, however, requires the user to specify the discipline desired as well as the date.

The FTR report is generated by combining the faculty ID and year, which are provided by the Educational Technician, with the data stored in the Class Instructor and Faculty objects. This report is used to show which courses an individual faculty member will be teaching during each quarter of the given year.

C. FEASIBILITY OF THE PROPOSED NEW SYSTEM

1. General

The Instruction Scheduling Program is a microcomputer-based decision support system which provides man-year estimation, instruction scheduling, report generation and database management capabilities. It draws heavily on a database management system for data retrieval and management of inputs and outputs. It is a single user system designed for stand-alone machines.

2. Cost

The cost associated with this project is minimal. All equipment and software that is needed to implement the system is already on hand. The time and effort required to train personnel on the system should be negligible. The Department Chairman, Associate Chair for Instruction and Educational Technician have a working knowledge of, and experience with, the microcomputers and their associated peripherals as well as the database management software available to the department. The simple users' manual (Appendix L) will also reduce the training effort.

3. Technical

A database management system is required which can handle a 150 kilobyte database with an annual growth rate estimated at 28% (Appendix J). It must be capable of functioning in a microcomputer environment and allow for ad-hoc query development. It must also allow multiple file

search and link capability. For efficient operation an 80286 based, or comparable, computer would be a minimum requirement, however, a 80386 or higher would be desirable.

4. Schedule

Development effort and development time for the Instruction Scheduling Program were estimated using the Constructive Cost Model (COCOMO) (Appendix I). Estimated Deliverable Source Instructions (EDSI) for each of the modules was estimated to be:

1. Database management modules - 1500 EDSI
2. Produce reports modules - 1000 EDSI
3. Build a schedule module - 500 EDSI

A total development time of 4.19 months was obtained as well as a total effort of 3.89 man-months. Given the five month time frame available to the developers, scheduling requirements were deemed feasible.

5. Recommendation

The Instruction Scheduling Program should be developed. It is financially, operationally and technically possible and practical. The projected cost will be minimal. The necessary hardware and software is available and the system can be implemented in the time available to the developers.

During the analysis phase, data that needs to be stored in the database and functions the system must satisfy are identified, and the feasibility of the proposed system is

assessed. When the analysis phase is complete, the results must be reviewed by the users. After the results have been approved, the process of system design can begin.

III. SYSTEM DESIGN

A. MANAGMENT INFORMATION SYSTEMS VS DECISION SUPPORT SYSTEMS

Both Management Information Systems (MIS) and Decision Support Systems (DSS) rely upon mechanisms for managing data. However, they differ in respect to the purpose for which the data are used. Basic database management applications and MIS typically use historical data for repetitive, routine transactions and report generation. On the other hand, a DSS uses models to transform data into information which can be used in a manager's decision making process. A DSS becomes a tool used by management to model some future states of their organization based on assumptions supplied by management. Since the Instruction Scheduling Program is concerned with extending the accounting and reporting functions to support the management decision making process, designing the program within the framework of DSS is appropriate.

B. DSS CHARACTERISTICS

Decision support systems go beyond the capabilities of a typical management information system by taking a broad view of the organization in terms of supporting and improving decisions. The focus of a DSS is not limited to semi-structured or unstructured decisions. DSS is a model-based system which, based on the needs of the problem being solved, uses one or more mathematical and/or statistical models to

assist the decision maker in evaluating alternative solutions. Contents of the database must go beyond just providing historical information about current and past operations. It must also contain appropriate external information.

Since a DSS is designed to enhance the decision processes of managers usually faced with ill-structured decisions, we do not always completely understand the user's requirements. As a result, we explicitly acknowledge that as part of our design and implementation effort the user will "learn" about the problem and, thereby, identify new information needs. Prototyping essentially bypasses information requirements definition by evolving requirements via the user learning process. It assumes requirements are only partially known at the start. (Ginzberg, Reitman, Stohr 1981:p. 80-81) DSS is, therefore, very amenable to the prototyping design strategy.

An effective DSS is easy to use, that is, not only does it assist the decision maker in supporting decisions via a man-machine interface, but also allows the person to address a problem using his/her own problem solving techniques.

C. DESIGN PHASE

Several factors critical for a successful system implementation were considered in designing the Instruction Scheduling Program. Some of the more important ones are listed below:

1. The system should be easily understood by both the manager/decision maker and the users.

2. The system should consider historical data on student population as well as current trends in schedule selection by students.

3. The system's parameters must be variable to permit evaluation of different assumptions.

4. Users will usually prefer a system that requires minimal maintenance and updating.

5. The entire system must be menu driven so that the users are not required to participate in a lengthy training program in order to learn it.

6. The system must be able to handle anticipated loads on currently available hardware (Zenith 248/IBM AT's).

It is during the design phase that the foundation of the database structure for the system is built. Throughout the design of this system "Deft CASE" from Deft Inc., Toronto, Canada, an automated computer-aided software engineering (CASE) tool, was used to improve the accuracy and cohesiveness of the diagrams, specifications, and structures. The use of the CASE tool also reduced the number of hours needed to proof and cross reference all the elements within the Instruction Scheduling Program, thus ensuring consistency.

D. LOGICAL DATABASE DESIGN

The design phase consists of two elements: the logical database design and the application design. During logical database design a blueprint of the database needed to support all applications should be produced (Kroenke, Dolan, 1988:p. 167). This is accomplished by examining the entities and identifying the relationships among them.

1. Object Oriented Approach

For this project an entity-relationship diagram (ERD) was developed (Appendix A) as well as an object diagram (Appendix C). The object diagram was further analyzed to identify relationships, and the objects were then transformed into relations. The rules of normalization were applied to those relations and, where anomalies were found, the design was modified in order to eliminate them.

a. Objects

In object oriented database design there are five different types of objects : simple objects, composite objects, compound objects, association objects and aggregation objects (Kroenke, Dolan, 1988:p. 212).

A simple object is the most basic of the five. It contains only single-valued, non-object properties. It can be represented by a single relation.

A composite object is one that contains one or more non-object multivalued properties and requires more than one relation in their representation.

A compound object contains at least one object property and will be represented by at least two relations, one for each object. The Course object and Curriculum object (Appendix C) are examples of a composite object.

An association object documents a relationship between two or more other objects. It also contains non-key

data. The Course Curriculum object (Appendix C) is an example of an association object.

The final object is an aggregation object which represents an entity group, that is, it represents a group of persons or things. The Class Instructor object (Appendix C) is an example of an aggregation object.

b. Relations

Data within the Instruction Scheduling Program are organized and stored in two-dimensional tables called relations. A relation can be thought of as a file with each row (tuple) of the relation being a record. Each record contains specific data items which are the attributes of the record. The goal of logical database design is to represent objects in the database using relations that (1) provide the data needed to construct user objects and (2) are robust enough to allow rows to be inserted, deleted, and modified without resulting in inconsistencies or errors in the stored data (Kroenke, Dolan 1988:p. 133). The relations created for the Instruction Scheduling Program are depicted in Appendix D.

2. Relation Diagram

a. Relationships

Binary relationships are the main building blocks for constructing objects. A binary relationship is a relationship that involves only two record types. Whereas an object is converted on a one-to-one basis into a relation, the

relationships between record types are not necessarily limited to one-to-one.

There are three types of binary relationships: one-to-one, one-to-many and many-to-many. The simplest form of binary relationship is one-to-one, that is, a record of one type is related to no more than one record of another type. There are no one-to-one relationships in the Instruction Scheduling Program. In the one-to-many relationship a record of one type is related to potentially many records of another type. All relationships in the Instruction Scheduling Program are one-to-many. An example would be the faculty/faculty expertise relationship where a single faculty member can conceivably have many areas (courses) of expertise. In a many-to-many relationship, a record of one type corresponds to many records of the second type and a record of the second type corresponds to many records of the first type. Since all the relations in the Instruction Scheduling Program are in third normal form (normalization is discussed in section D.3 of this chapter) there are no many-to-many relationships in the Instruction Scheduling Program. (Kroenke, Dolan, 1988:p. 168-178) The completed relation diagram for the Instruction Scheduling Program is found in Appendix D.

b. Constraints

Another notation used in Appendix D is one to indicate a mandatory or optional relationship between two record types. A circle at the end of a line indicates an

optional association. A bar at the end of a line indicates a mandatory association. An example of an optional association would be between Faculty and Class Instructor. A faculty member may be a class instructor also. An example of a mandatory association would be between Course and Discipline. A course must have a discipline associated with it. Class and Class Instructor is an example of a mandatory association in both directions. A class must have a class instructor assigned to it and a class instructor must have a class to instruct.

c. Relation Keys

Each relation has a set of attributes, called the key, that uniquely identifies each record. These keys are identified in the relation diagram (Appendix D) as underlined attributes in each relation. In the Faculty relation, Faculty ID is the key which uniquely identifies each record. In the Faculty Availability relation, Faculty ID, Quarter and Year are all required in order to uniquely identify each record in the relation. Since Faculty ID is the key for a different relation, specifically Faculty, it is referred to as a foreign key within Faculty Availability. Foreign keys are identified in the relation diagram by an asterisk.

3. Normalization

Normalization is the process used to develop well structured, robust relations. There are seven different normal forms that a relation can take. As a relation

progresses through the different forms, the different types of modification anomalies that it is vulnerable to are removed. Only the Domain/Key Normal Form guarantees the relation will have no anomalies. A relation is in DK/NF if every constraint on the relation is a logical consequence of the definition of keys and domains. The essence of DK/NF is that every relation must have a single theme. (Kroenke, Dolan, 1988:p. 133-153) The relations developed for this project are all in at least third normal form (3NF). A relation can be considered in 3NF if (1) all non-key attributes are dependent on all of the key, and (2) if it contains no transitive dependencies. For example in the Course Curriculum relation the non-key attributes, Quarter Taken and Course Curriculum Percentage, are dependent on the whole key, that is, on both Course Number and Curriculum Number. Additionally, there are no transitive dependencies within the relation.

4. Data Integrity

Data integrity deals with the problem of ensuring the data in the database are accurate and/or valid. Inconsistency between two entries that are suppose to represent the same "fact" is an example of lack of integrity. Data integrity is even more important in a multi-user database system than in a "private file" environment such as the Instruction Scheduling Program. (Date 1990:p. 16)

Maintaining data integrity in the Instruction Scheduling Program is accomplished through the enforcement of

data constraints via procedural code. The procedural code ensures appropriate constraints are used and creates procedures to ensure data integrity if the constraints are violated. Two integrity rules, as defined by Date(1990), were helpful in designing the database structure; the Entity Integrity Rule and Referential Integrity Rule.

a. Entity Integrity Rule

This rule states simply that the primary key in base relations is not allowed to accept nulls, or blanks. This rule is satisfied in the Instruction Scheduling Program through validation procedures coded into the program. An example is the Curriculum relation. Curriculum Number is the primary key and is validated when entered to ensure the curriculum exists if being modified or deleted, or does not exist if being added. Data input constraints must also be met for Curriculum Numbers being added. If these validation procedures are not in place, records can be created or modified that are not uniquely identifiable and cannot be linked to related files. Inaccuracies are injected into the database that are difficult to identify and correct. For example, if the Curriculum Number were allowed to be null, data in the Course and Course Curriculum relations could not be uniquely identified or related with valid curricula. In addition, the user would be unable to associate a section, found in the Section relation, with its parent curriculum.

b. Referential Integrity Rule

This rule states the database must not contain any unmatched foreign key values, that is, a foreign key (that is not blank) for which there is no matching value of the primary key in the target relation (Date 1990:p. 284-285). It is very specific in that the foreign key must match the primary key, not alternate keys. This rule is satisfied in the Instruction Scheduling Program through validation procedures and by not allowing foreign keys to accept blanks. In the Section relation, Curriculum Number is a foreign key. The target relation for Section is Curriculum where Curriculum Number is the primary key. Curriculum Number is not allowed to be blank in either relation, and through the validation process must always form a match.

E. APPLICATION DESIGN

An application extracts the appropriate data from the database and presents it in a predefined format to the user. The object-oriented approach to application design consists of four steps: Determine applications and scope, design control mechanisms, determine options for each menu, and design the format for data presentation (Kroenke, Dolan, 1988:p. 264).

1. Applications and Scope

During the requirements phase, the functionality requirements for this project were determined and grouped into modules for development. Each module was decomposed into sub-modules that ensured all necessary functionalities identified

by the users were present and capable of producing the desired results. These modules were then transposed into a hierarchy structure as shown in Appendix E. This hierarchy structure serves to depict the applications and scope of this project.

2. Control Mechanisms

In this step of the design process the decision was made to control the applications through a menu driven mechanism as opposed to a command driven mechanism. Menus are largely self-explanatory, require less training to use and, therefore, are generally easier to use than commands. This also allows the system to perform a logical sequence of functions and subfunctions with each logical sequence generating a path of actions.

3. Determine Options for Each Menu

After completing the requirements phase and the applications design phase, the decision was made to use an action/object menu design hierarchy. With this type of structure the user starts with a list of actions and is then led to lower level menus where objects can be chosen on which to perform the action. Examples of the various menus that correspond to the menu hierarchy are found in Appendix G.

4. Design the Format for Data Presentation

The decision on data presentation format was made by the Associate Chair for Instruction and his assistant during the initial interviews with them. That decision was to keep all output and reports in the same format as the present

system. The primary reason for the decision was to reduce the amount of training time required for the department to learn the new system and to interpret the data. Output and report formats are found in Appendix H.

During the design phase, the database for supporting all applications was developed as was the presentation format for the applications. With the design phase completed, the process of physically constructing and implementing the system can begin.

IV. IMPLEMENTATION

A. GENERAL

System implementation includes all those activities that take place to convert the old system to the new one. The main task of implementation is to construct a system which meets the design specifications. Proper implementation adapts the system analysis and design to provide a reliable system that meets the organization's requirements (Senn, 1984:p. 525).

1. Software Selection

dBASE IV was selected by the Administrative Sciences Department as the program of choice for their automated database system. dBASE IV is available throughout the department and the users of the Instruction Scheduling Program have experience in using dBASE IV.

dBASE IV is a relational database management system which features the choice of a completely menu-driven interface (Control Center) for creating and manipulating data structures and a powerful database language. dBASE IV offers assistance to users through a Control Center or through the built-in help system which provides reminders of the syntax and options of all commands and functions.

Utilizing the control center within dBASE IV allows even the computer novice to accomplish many database management tasks which could have only been accomplished by

experienced computer programmers in the not-too-distant past. Creating and manipulating database structures, creating screens and reports and creating queries are but a few examples of those tasks made easy by use of the Control Center.

The dBASE IV programming language consists of over 360 commands and functions with additional system memory variables and database configuration commands (Simpson, 1989:p. xxii). This programming language allows the experienced programmer the ability to add flexibility to the system being developed as well as the ability to produce more sophisticated applications. The dBASE IV programming language was used exclusively in developing this system. It will adequately support the Instruction Scheduling Program requirements, and is presently installed on the Educational Technician's computer.

The relations, records and attributes developed during the design phase are translated into database specific files, records and fields during implementation (APPENDIX F). Based on the application design, and the requirement to utilize dBASE IV, the application development tools within dBASE IV were used to construct the DBMS tables. The Computer-aided Software Engineering (CASE) tool "Deft CASE" was used to prototype the initial forms, reports and menus, however, once these designs were approved, the final forms were developed using the tools available in dBASE IV.

2. Hardware Selection

The minimum hardware requirements for dBASE IV are:

1. An IBM PC/XT, AT, or PS/2 model 30, 50, 60, or 80; or any 100 percent compatible micro-computer.

2. IBM DOS or MS-DOS version 2.0 or higher for the DOS version of dBASE IV.

3. At least 640K of RAM.

4. A hard disk with about 3.5 megabytes of available disk space. (Simpson, 1989:p. 762)

Hardware currently available to the Educational Technician consists of an IBM AT computer with a 30 megabyte hard drive, and a Hewlett Packard LaserJet III printer. This will adequately support the Instruction Scheduling Program although a 386-based machine is more desirable.

B. SOFTWARE DOCUMENTATION

1. User Manual

The user manual for this system is provided in Appendix L. It is designed for the user who is familiar with dBASE IV but does not require a thorough knowledge of dBASE IV or the hardware that the system is running on (basic familiarity with DOS is assumed). The manual includes the step by step procedures necessary to realize the developed functionalities and includes a description of all menus and reports the user may encounter while using the system. Also included in the manual are the various field constraints, formats and masks which must be adhered to while manipulating the database or the system outputs.

2. Main Program

dBASE IV allows the programmer two options for programming. The first option utilizes the built in Control Center in dBASE IV, as discussed previously, whereas the second option utilizes the dBASE IV programming language and user entered commands. The program developed for this project is written in the dBASE IV programming language. We chose to use the programming language because of the additional flexibility user entered commands provide compared to the command center. The dBASE IV programming language proved to be an adequate programming environment for this project. All critical functionalities were achieved using constructs available within the language.

The main program consists of more than 65 separate modules which are grouped into four categories (Appendix K). The first category deals with maintaining the database and includes all of the Add, Modify, Delete and List Modules. The second category deals with producing written reports and includes all of the Print Modules. The third category deals with the building of the different schedules and reports. This third category includes the Calculate Enrollment, Determine Instructor, Assign and Build Annual Modules. The fourth category deals with estimating man-year requirements. This category includes the Calculate EMY, What-If, Save What-If and Load What-If Modules. It also allows for printing of the "what-if" report.

By designing each of the modules with total system integration in mind from the beginning, the final implementation of each of the modules was a smooth process. The modules were composed using structured programming techniques (Gilbert, 1983:p. 406) in order to ensure the program text could be easily understood, maintained and modified where necessary. This is important because the program may be corrected or modified many times in the future by different people unfamiliar with it. There are few intricacies laced through the program so that maintenance personnel can quickly discover which portions of the program require needed maintenance or modification.

Each module was refined so that it would embody only a particular functionality and all details pertinent to that design decision or functionality are found in the associated module. All the modules are highly cohesive and loosely coupled. Simply put, the lines of code that work together in the program also appear together in each of the modules.

C. REPORTS

The user has the option of producing several reports (Appendix H) at various levels of aggregation with this system. The format of the reports was unchanged from the previous system, with the exception of those additional reports required for this project. All the reports were generated using the dBASE IV programming language. A brief description of the reports follows:

1. Teaching Schedule by Discipline (TSD)

This report is the starting point for the generation of the other reports within this system. It provides the department with a comprehensive listing of which instructors will be teaching what courses during each quarter of a given academic year. The courses are listed under their appropriate discipline, in accordance with their mandatory relationship established earlier.

2. Quarterly Teaching Schedule (QTS)

This report provides the Associate Chair for Instruction a breakdown of the teaching assignments within the department by course for any given quarter. Unlike the TSD report, the courses are not broken down by discipline and an additional element of data, the number of students enrolled in each course, is added. This report allows the Associate Chair for Instruction to concentrate on "fine tuning" a given quarter while working out scheduling exceptions.

3. Course Offerings (CO)

This report is simply a listing of all courses that are taught by the Administrative Sciences Department that will be offered during the specified quarter. This report is sent out to all the other departments at the school for purposes of student course selection. This report is significant in that, if it is properly used by the other departments, it will prevent students from selecting courses that will not be offered by the Administrative Sciences Department. This in

turn will prevent last minute schedule changes by students at the beginning of each quarter.

4. Estimated Man-Year (EMY)

This report is actually provided to the user in two different forms for two separate purposes. The first form is a printed report that provides the estimated faculty man-year requirements for an academic year, that is, how many faculty are required in the Administrative Sciences Department to teach the courses that are required for that year. The information is broken down by discipline and by quarter to assist the department chairman in analyzing the data. In its final form this data can be used by the department in making financial plans for future academic years and quarters.

The second form is an on-line management decision aid focused specifically on the man-year requirements issue within the department. With this second form either the Department Chairman or the Associate Chair for Instruction is able to make adjustments to the estimated student enrollment figures used in determining future man-year requirements.

The basis of the man-year estimate calculations is the Course Curriculum Percentage which is an attribute of the Course Curriculum object. This percentage is derived from historical class enrollment data obtained from the Registrar's Office, and curriculum enrollment figures and recommended course schedules from the curriculum offices. The number of students from each section of the curriculum that enrolled in

a specified class during a specified quarter, as identified in their curriculum course matrix, was summed and compared to the total number of students in their respective curriculum.

The percentage obtained from that comparison represents the percentage of students from that curriculum who followed their curriculum course matrix in choosing when to take courses. An example of the percentage calculations is found in Figure 1. That percentage is then applied to anticipated curriculum enrollment figures to forecast the number of students from that curriculum who will enroll in classes as specified by their course matrix. If courses are not offered during the quarter specified by a curriculum's matrix, the number of students in the affected section are not included in the class or curriculum totals when figuring the percentage. This prevents course scheduling problems/changes from affecting the percentages. An example would be Curriculum 827, section MM74 in Figure 1, which is not included in the curriculum total of 63 for this reason.

Additionally, courses that are validated by students and courses that are offered as electives are not addressed in these calculations.

Curriculum 827

Start date:	0187	0787	0188	0788	0189	0789
Sec. Name :	MM72	MM74	MM82	MM84	MM92	MM94
Sec. Size :	7	15	11	16	7	22

Class (taken in first quarter)

MN2150	11	9	20	4	17
MN2031	11	9	18	4	17
MN3333	11	9	17	10	17

	Curr. Tot.	Class Tot.	Percentage
MN2150	63	61	0.968
MN2031	63	59	0.937
MN3333	63	64	1.016

Figure 1. Percentage Calculations

Giving the decision makers within the department the ability to "what-if" the student enrollment figures allows them to analyze alternatives from "best case" to "worst case" estimates. This flexibility will allow the department to generate a more meaningful budget forecast which takes into account the inherent uncertainty of student enrollment data.

5. Estimated Teaching Schedule by Discipline (ETSD)

This report follows the same format as the TSD report except for the listing of instructors, which has been removed. It provides the Department Chairman and the Associate Chair for Instruction a comprehensive listing of which courses will be required during each quarter of a given academic year and the number of sections of that course that are required. This report is significantly different from the TSD, however, in that it is based on the "what-if" student enrollment figures

input by the chairman or associate chair during the man-year estimation process. It will allow a more detailed analysis of the alternatives explored in the "what-if" scenarios by addressing specific class needs that result from the student enrollment figures used.

6. Faculty Teaching Report (FTR)

This report is a listing of all the faculty in the Administrative Sciences Department who have teaching requirements during a given academic year. It lists the faculty in faculty ID order and displays the courses and quarters they will be required to teach. This report allows the department to determine teaching loads of individual faculty members without having to conduct an exhaustive search of the TSD.

7. Teaching Schedule for One Discipline (TSOD)

This report is similar to the TSD report. Whereas the TSD shows the information for every discipline in the department, this report only shows the information for one discipline at a time. It allows for analysis of the teaching schedule, as does the TSD, only in smaller pieces.

For any system implementation to be successful, the overall value of the system must be perceived by its users to be high enough to warrant their investment of time and effort to learn and then use it. The system must also be easy for inexperienced or casual users to operate, yet allow the more experienced user the flexibility in manipulating the system

using commands. Other areas that apply in assuring successful implementation are user training, user documentation and user assistance. They must be carefully planned and constantly reevaluated to ensure they meet the needs and requirements of the organization.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSION

The amount of data required to accurately forecast and schedule classes and faculty within the Administrative Sciences Department exceeded the manual data handling capabilities of the department. The inaccuracy and inconsistency of the data caused difficulty in obtaining accurate and timely reports and schedules. This project develops a decision support system and its associated relational database to assist the department in these critical activities. The main goal is to provide department planners with accurate forecasts of faculty requirements needed to meet anticipated instructional loads. The implemented system can potentially increase the productivity, effectiveness and flexibility of department personnel involved in the forecasting and scheduling processes.

Development efforts focused heavily on the analysis and design of the Instruction Scheduling Program. A total of five months development time and ten man-months development effort were expended to complete the program. Our projections of the development time and effort fell well short of the actual time and effort required, despite an initial, detailed COCOMO estimate (Appendix I, Figure 1). A subsequent COCOMO estimate (Appendix I, Figure 2), made four months after the first and

using more accurate data, still failed to reflect the actual time and effort required to complete the system. Compared to the most recent estimate, it still took an additional 5.11 months and 4.16 man-months of effort to complete the program. A contributing factor to the discrepancy is the developers' status as students. Full time commitment to the project during most of its development was impossible due to other academic requirements. In addition, much more time was needed for the coding effort than anticipated. This was due, in part, to both developers having to learn the dBASE IV programming language while trying to apply that new knowledge to a complex problem. In the end, good programming practices used in conjunction with a structured development methodology proved critical to the successful development of the program.

The Instruction Scheduling Program is viewed as four interactive applications. The database management application permits and facilitates the management and maintenance of the database files. The scheduling application generates teaching schedules based on user inputs. The report generating application produces all the reports required of the system by the department. The estimation application allows input and modification of student enrollment data used in the man-year estimation process. By using a menu driven structure the system is user friendly and requires a minimum amount of training to become proficient in its use. The user can easily

progress through the program applications with little prior knowledge on the use of dBASE IV.

dBase IV proved to be adequate for the programming needs of this project and provided all the functionality necessary to complete the Instruction Scheduling Program. The hardware available to the department also proved to be minimally adequate, however, the system should ideally run on a 80386-based, or higher, computer with a built-in system clock.

B. RECOMMENDATIONS

Establishment of a relational database allows for future expansion and capability enhancements. It also requires consistent and timely maintenance. Quarterly updating of all the files used in the Instruction Scheduling Program is critical. The curriculum offices that provide input to this system should be required to submit their data to the Educational Technician on a quarterly basis. The curriculum offices must also ensure the data they submit are accurate and complete. In addition, updating of the class enrollment data from the registrar's office should be conducted bi-annually.

Future enhancements that would increase the functionality and usability of the program are listed below:

1. To increase the accuracy of the system, include course electives offered by each curriculum and account for the students who do not follow their course matrix.

2. To improve the accessibility of the Instruction Scheduling Program, adapt the program to facilitate a multiuser environment. DBASE IV has the capability to be used in a network environment.

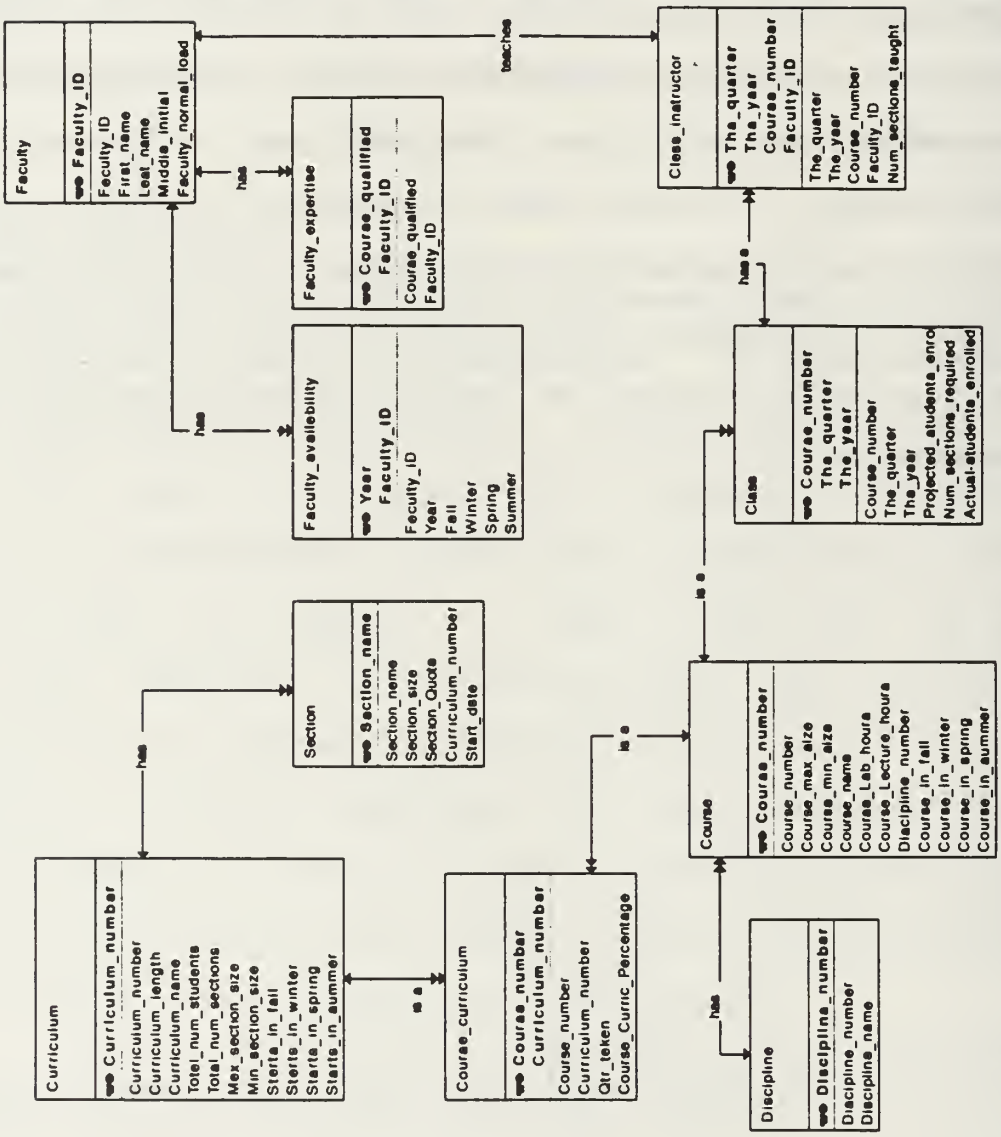
3. Incorporate the Faculty Availability and Faculty Expertise files into the scheduling application. This would allow their data to be available to decision makers prior to final schedule generation. Currently, they are used only for output to reports.

4. Optimize the source code to improve the system's efficiency and speed.

The Instruction Scheduling Program is a comprehensive automated decision aid designed for a microcomputer environment. It will assist department planners in their forecasting and scheduling activities by providing timely and accurate information. It will also improve the productivity of personnel involved in scheduling by automating report generation.

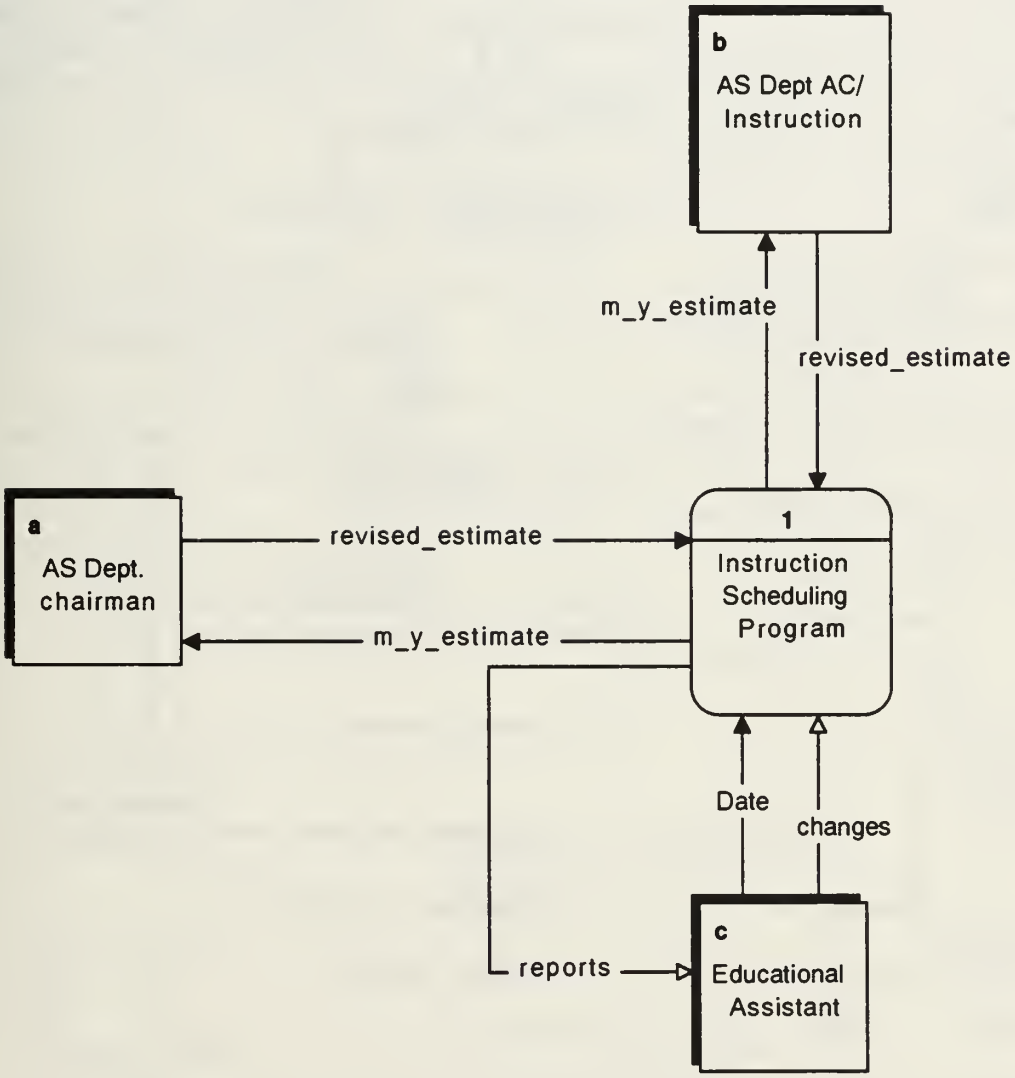
APPENDIX A

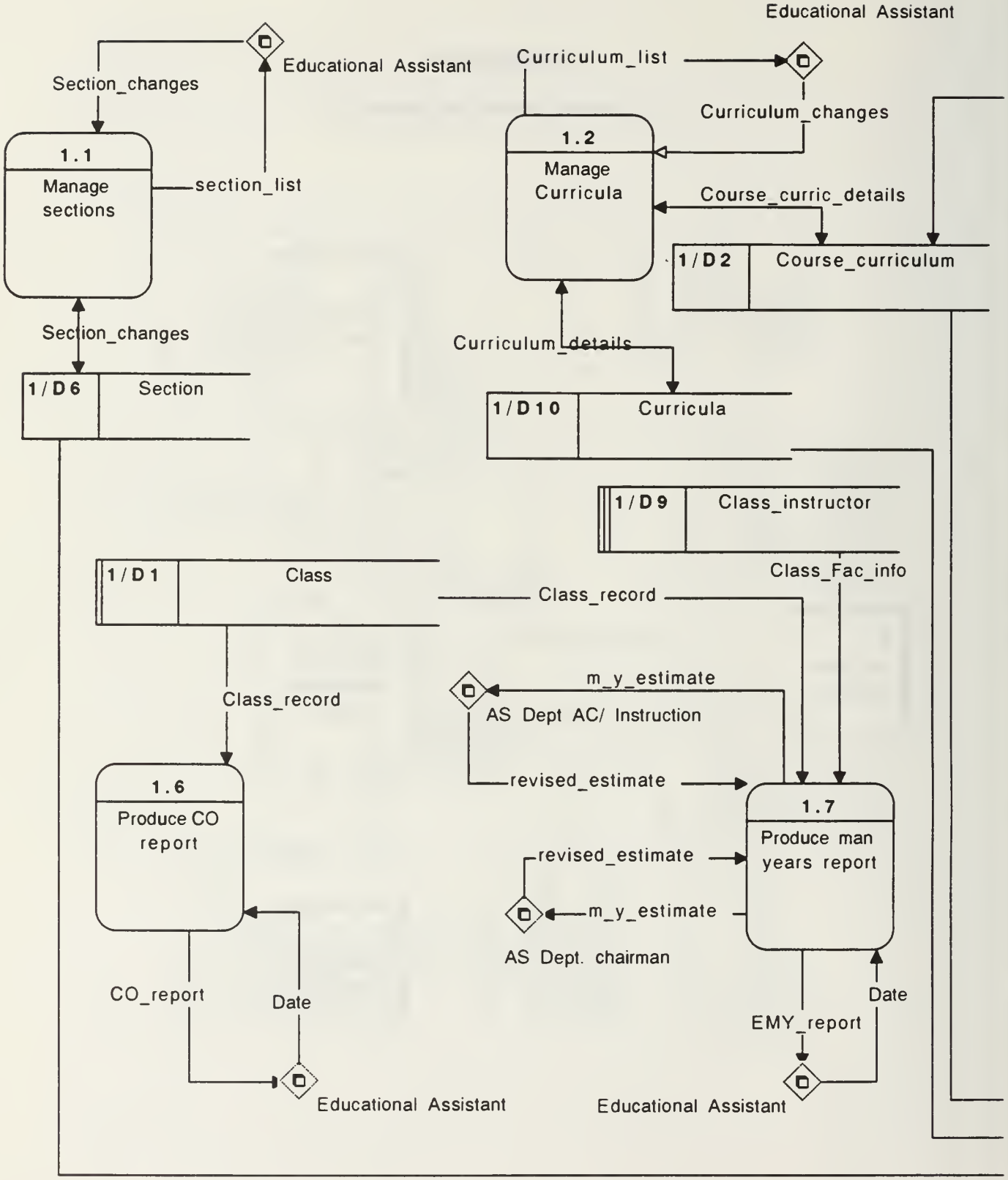
ENTITY RELATIONSHIP DIAGRAM

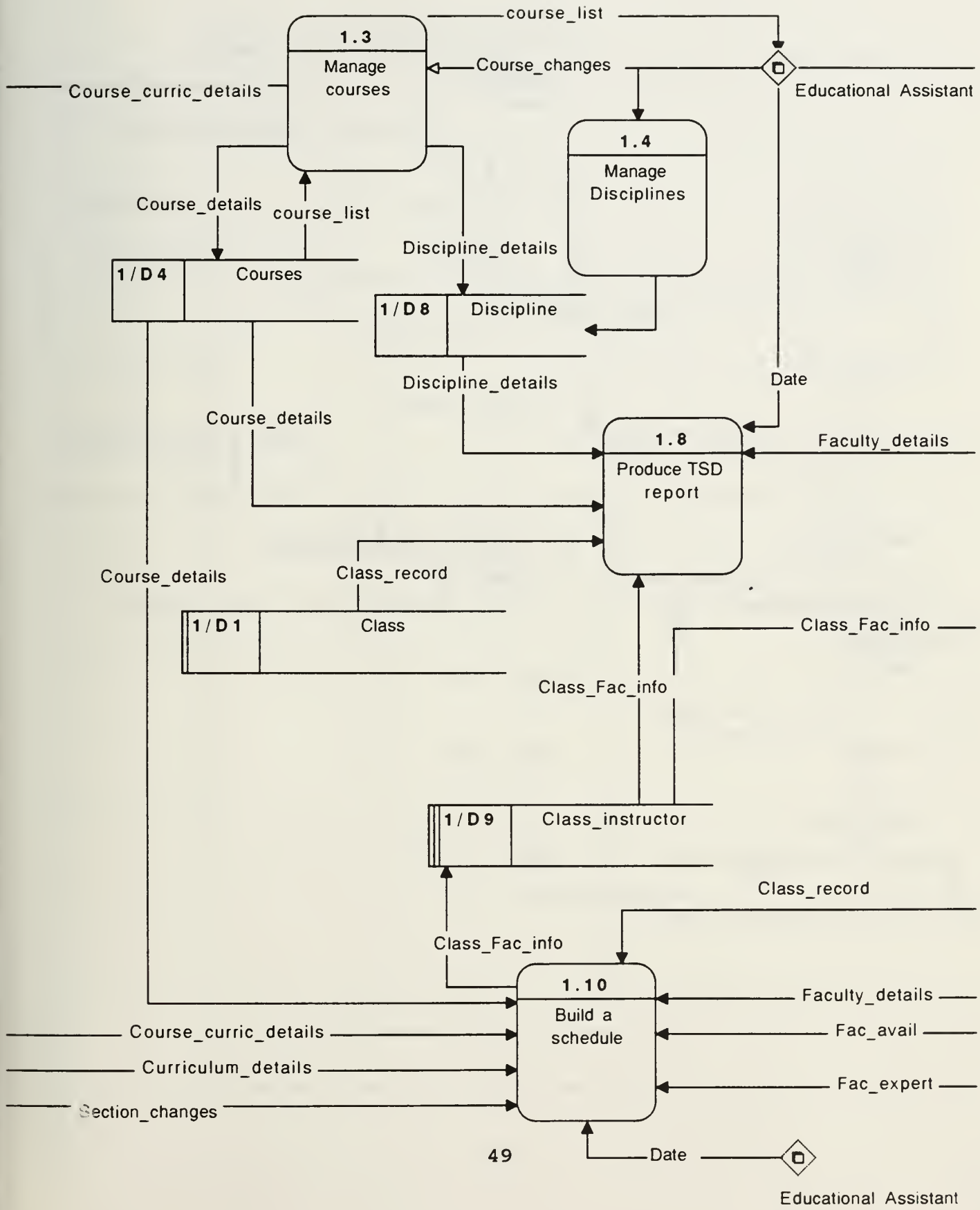


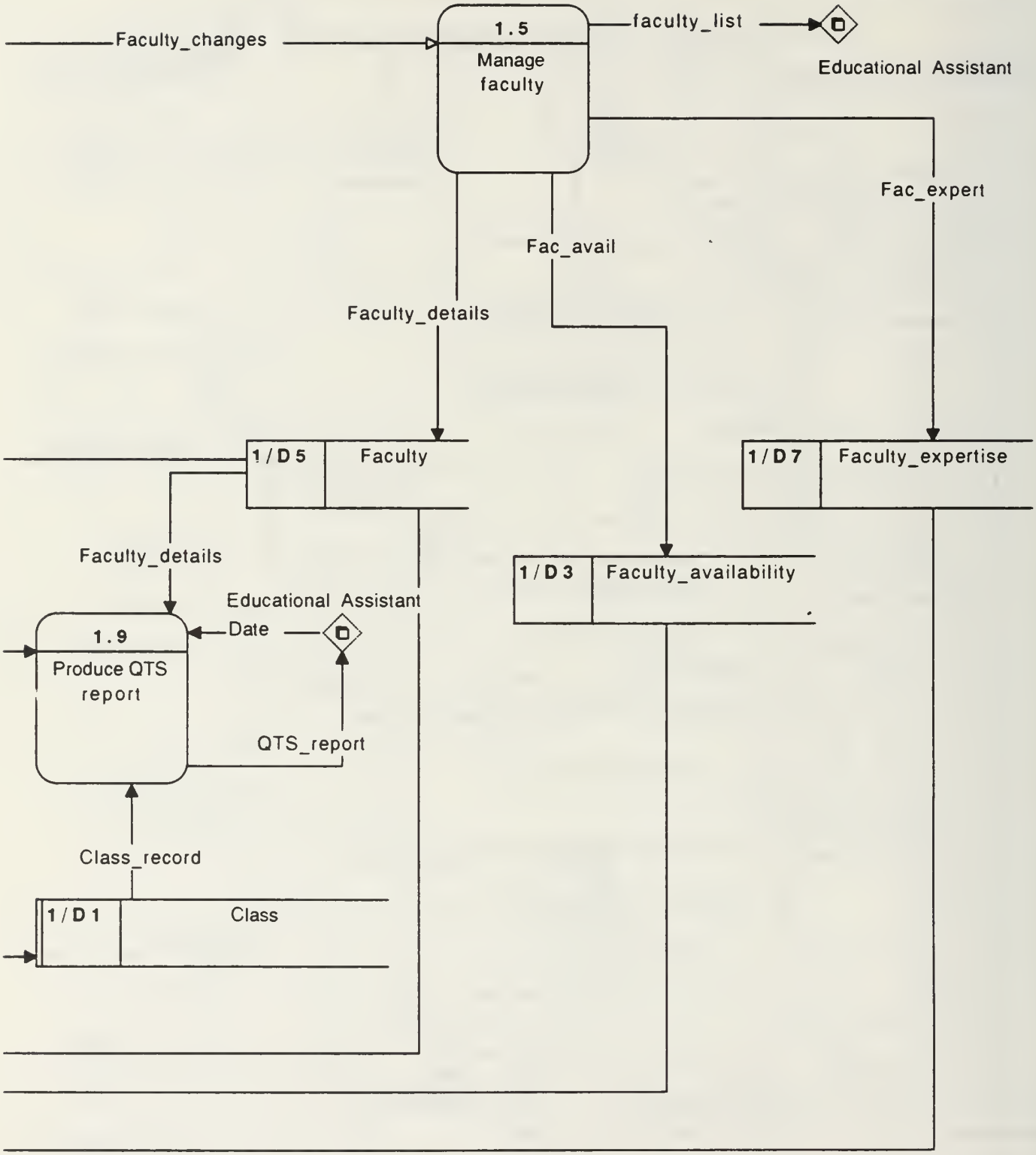
APPENDIX B

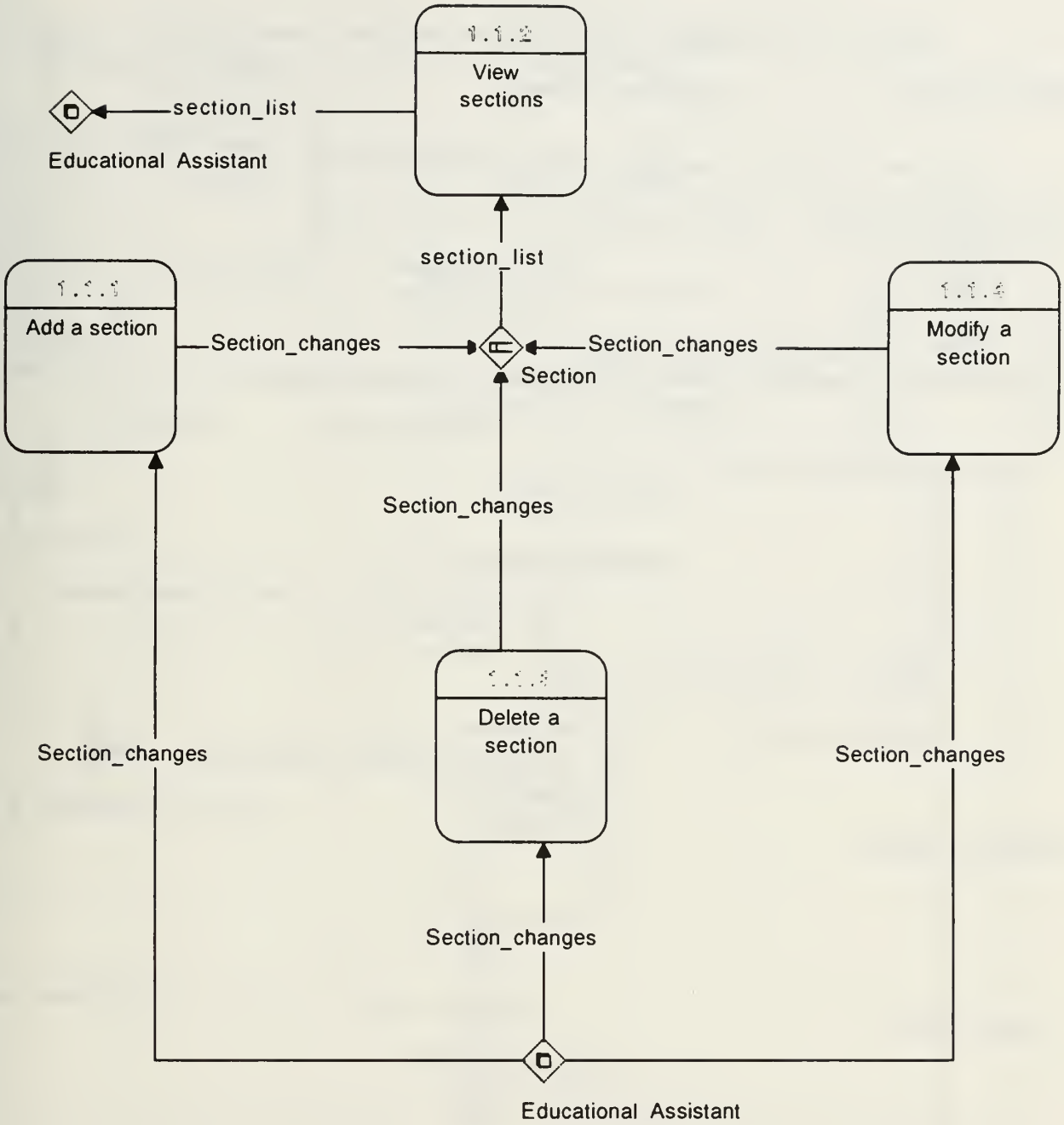
DATA FLOW DIAGRAMS

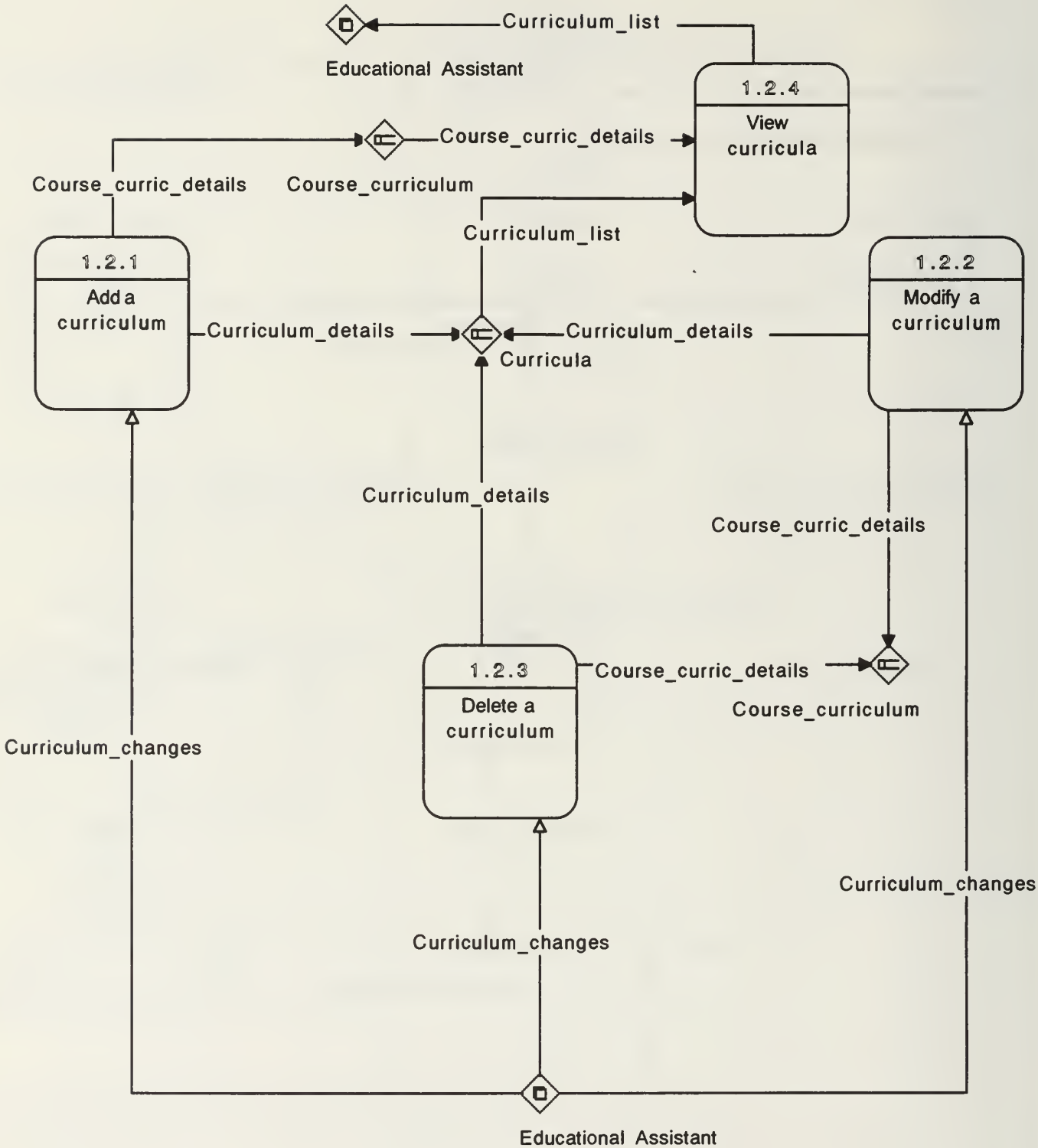


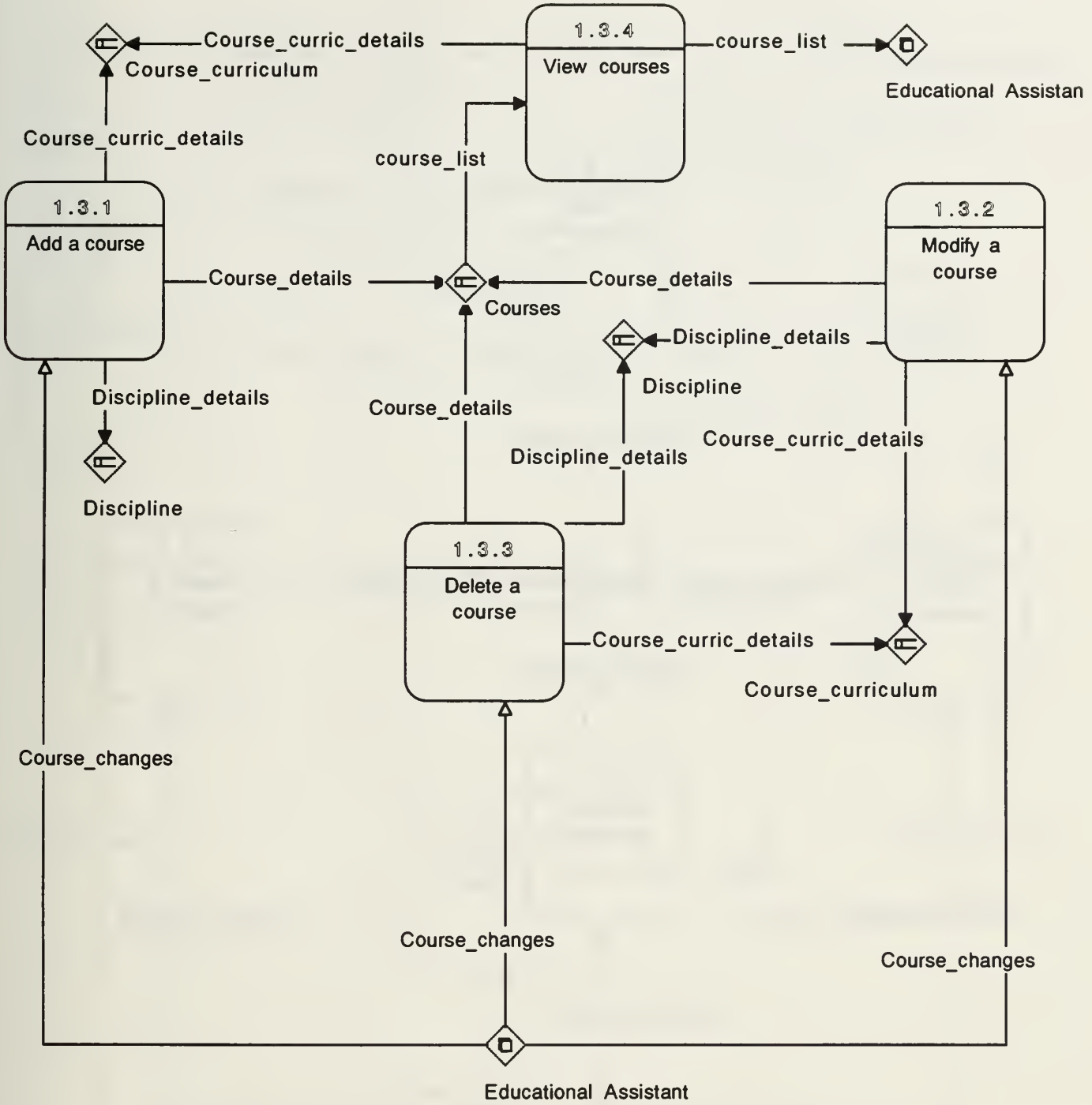

Educational Assistant
Section_changes
Educational Assistant
1.1
Manage sections
section_list
Section_changes
1/D6
Section
Curriculum_list
Curriculum_changes
1.2
Manage Curricula
Course_curric_details
1/D2
Course_curriculum
Curriculum_details
1/D10
Curricula
1/D9
Class_instructor
Class_Fac_info
1/D1
Class
Class_record
Class_record
m_y_estimate
AS Dept AC/ Instruction
revised_estimate
1.6
Produce CO report
1.7
Produce man years report
revised_estimate
m_y_estimate
AS Dept. chairman
CO_report
Date
Date
EMY_report
Educational Assistant
Educational Assistant

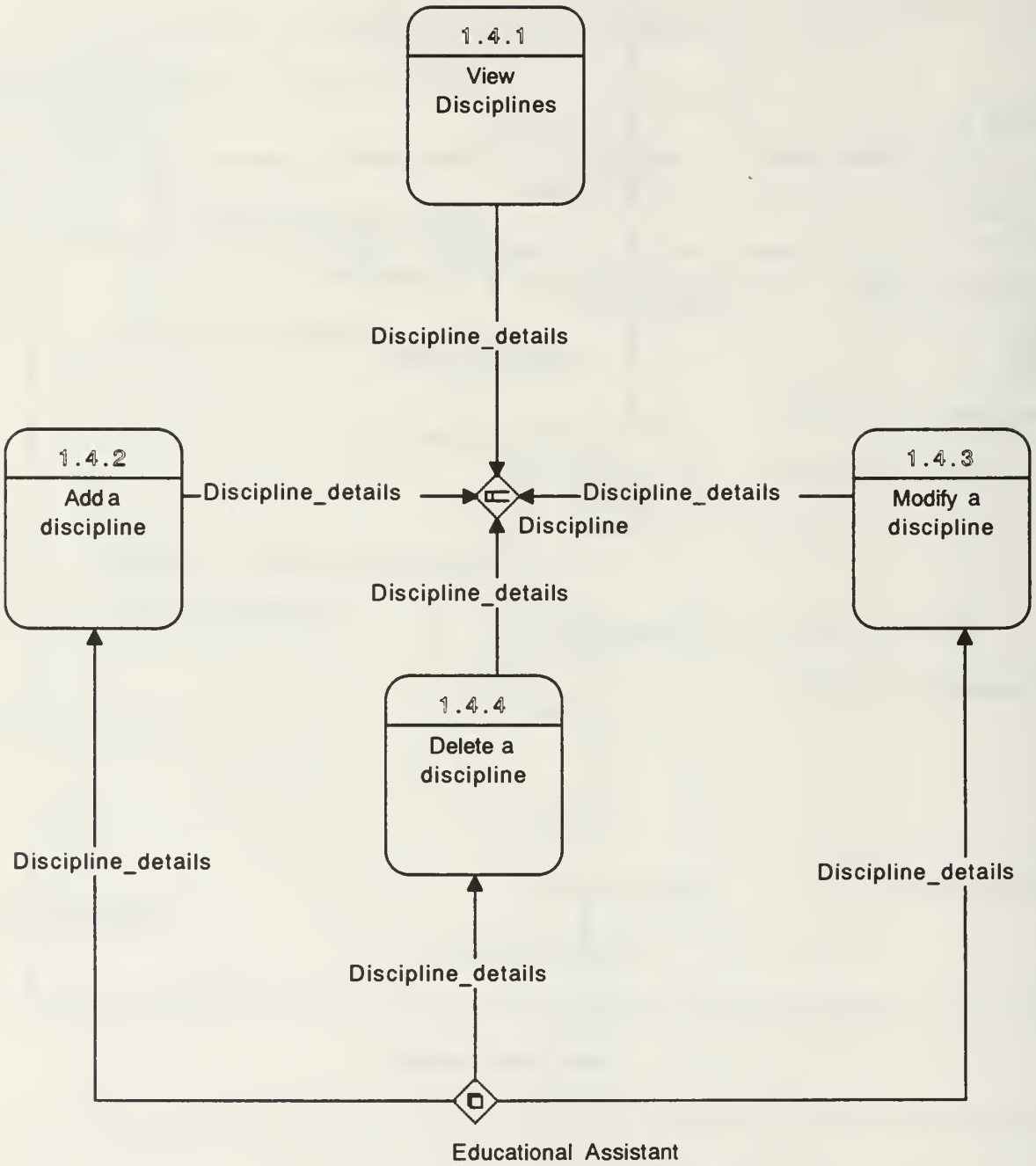

1.3
Manage courses
course_list
Course_changes
Course_curric_details
Educational Assistant
1.4
Manage Disciplines
Course_details
course_list
Discipline_details
1/D4
Courses
1/D8
Discipline
Discipline_details
Date
Course_details
1.8
Produce TSD report
Faculty_details
Course_details
Class_record
1/D1
Class
Class_Fac_info
Class_Fac_info
1/D9
Class_instructor
Class_record
Class_Fac_info
1.10
Build a schedule
Faculty_details
Course_curric_details
Fac_avail
Curriculum_details
Fac_expert
Section_changes
Date
Educational Assistant

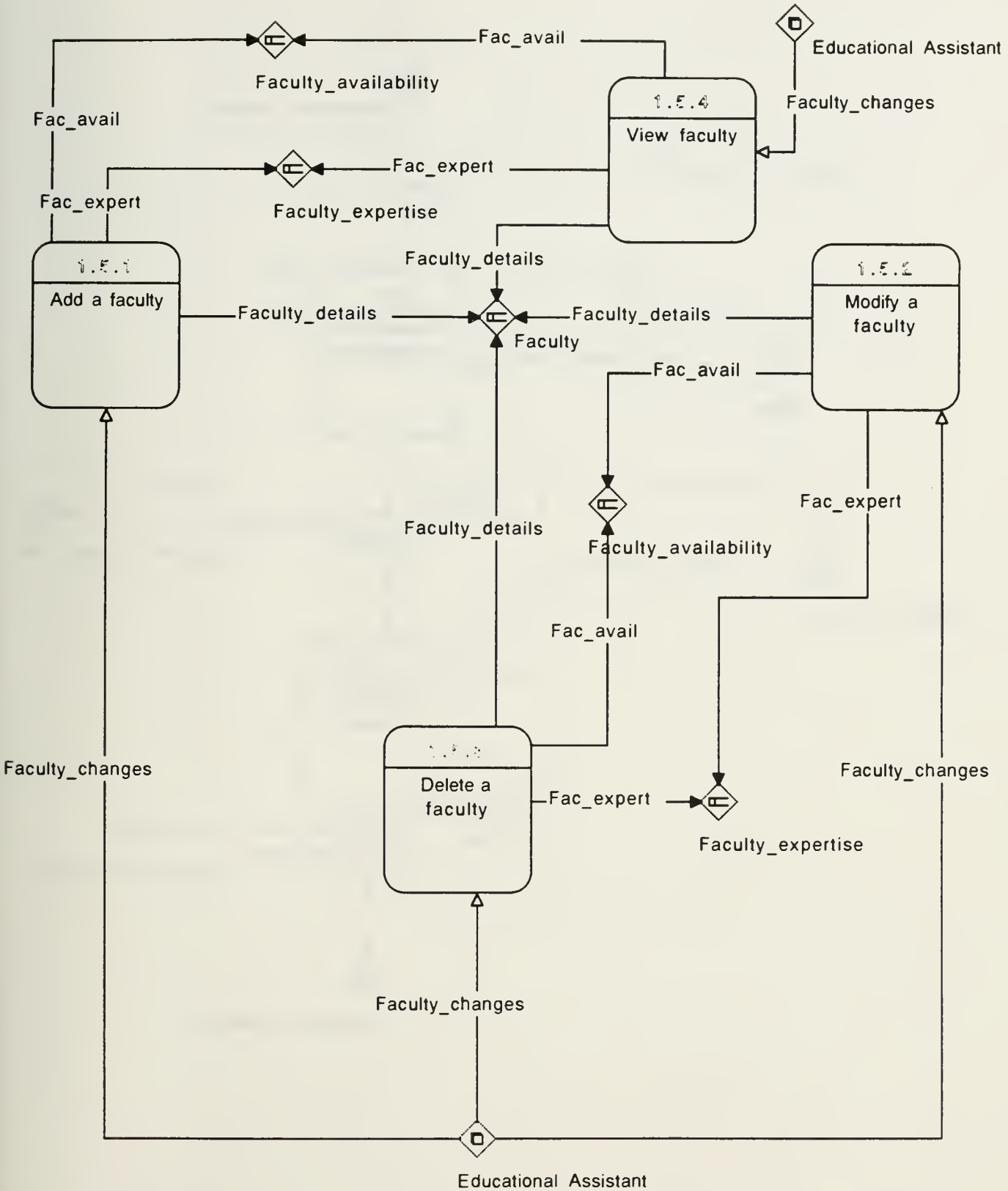

Faculty_changes
1.5
Manage faculty
faculty_list
Educational Assistant
Fac_expert
Fac_avail
Faculty_details
1/D5
Faculty
1/D7
Faculty_expertise
Faculty_details
1/D3
Faculty_availability
Educational Assistant
Date
1.9
Produce QTS report
QTS_report
Class_record
1/D1
Class

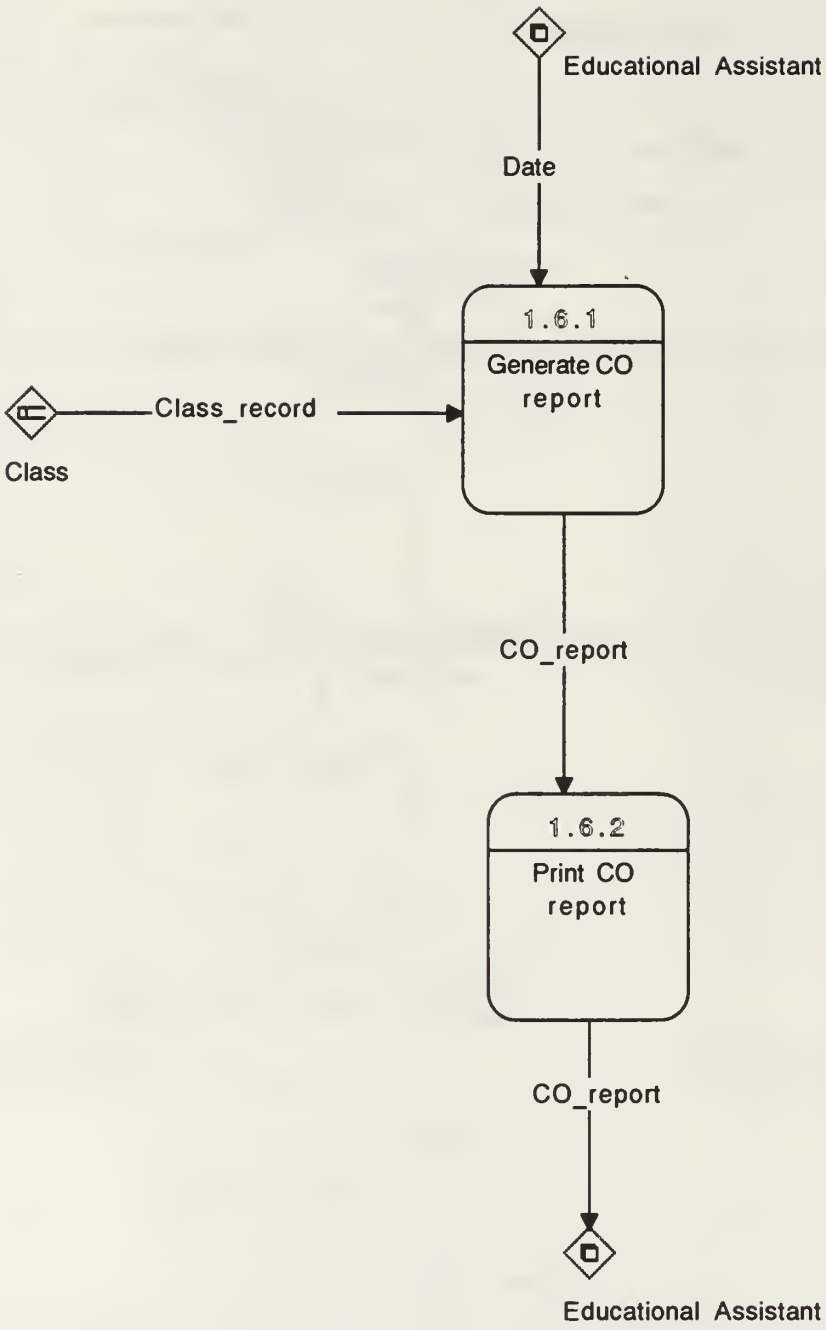

View sections
section_list
Educational Assistant
section_list
Add a section
Section_changes
Section_changes
Modify a section
Section
Section_changes
Delete a section
Section_changes
Section_changes
Section_changes
Educational Assistant

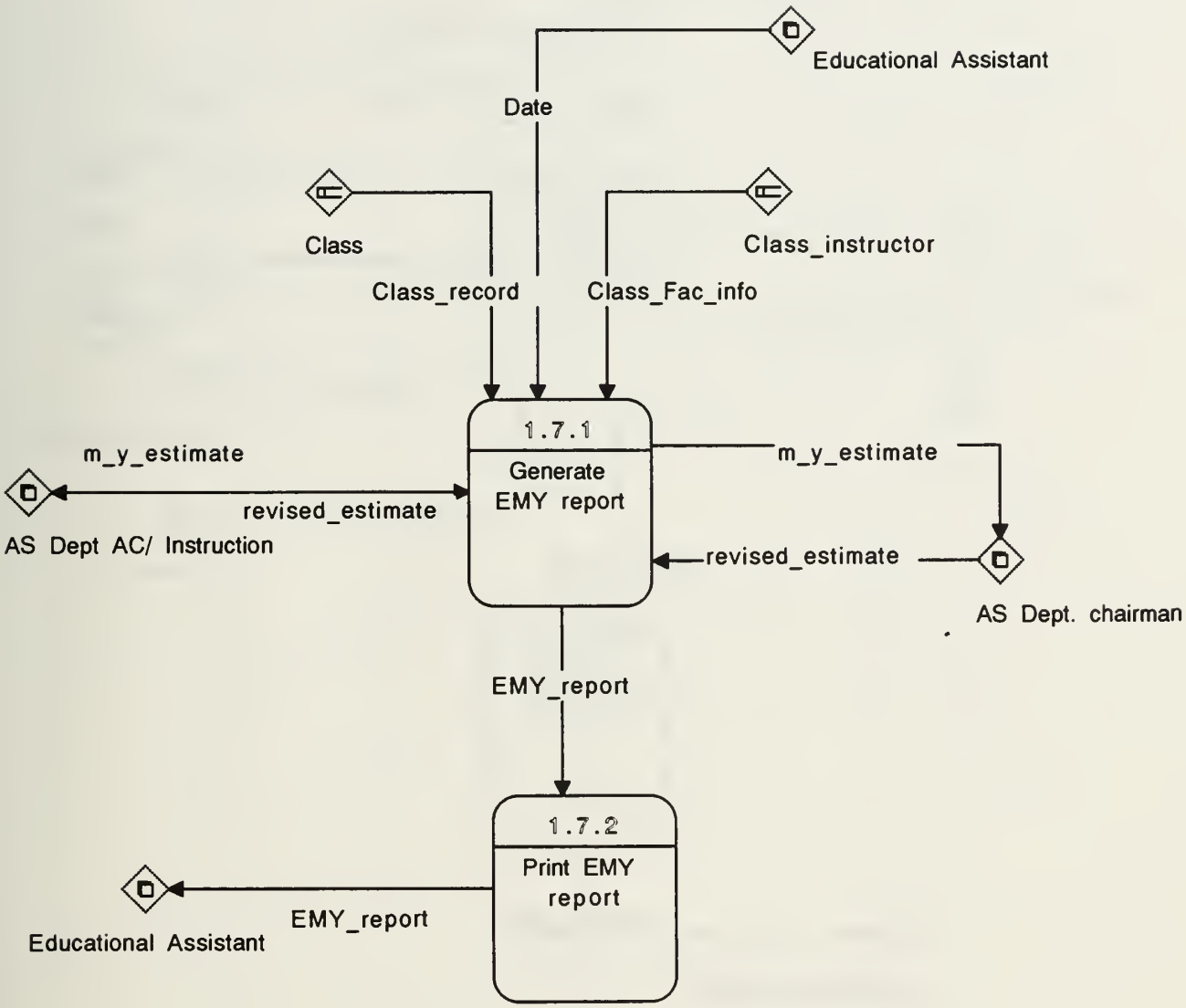

Curriculum_list
Educational Assistant
1.2.4
View curricula
Course_curric_details
Course_curric_details
Course_curriculum
Curriculum_list
1.2.1
Add a curriculum
Curriculum_details
Curriculum_details
Curricula
1.2.2
Modify a curriculum
Curriculum_details
Course_curric_details
1.2.3
Delete a curriculum
Course_curric_details
Course_curriculum
Curriculum_changes
Curriculum_changes
Curriculum_changes
Educational Assistant

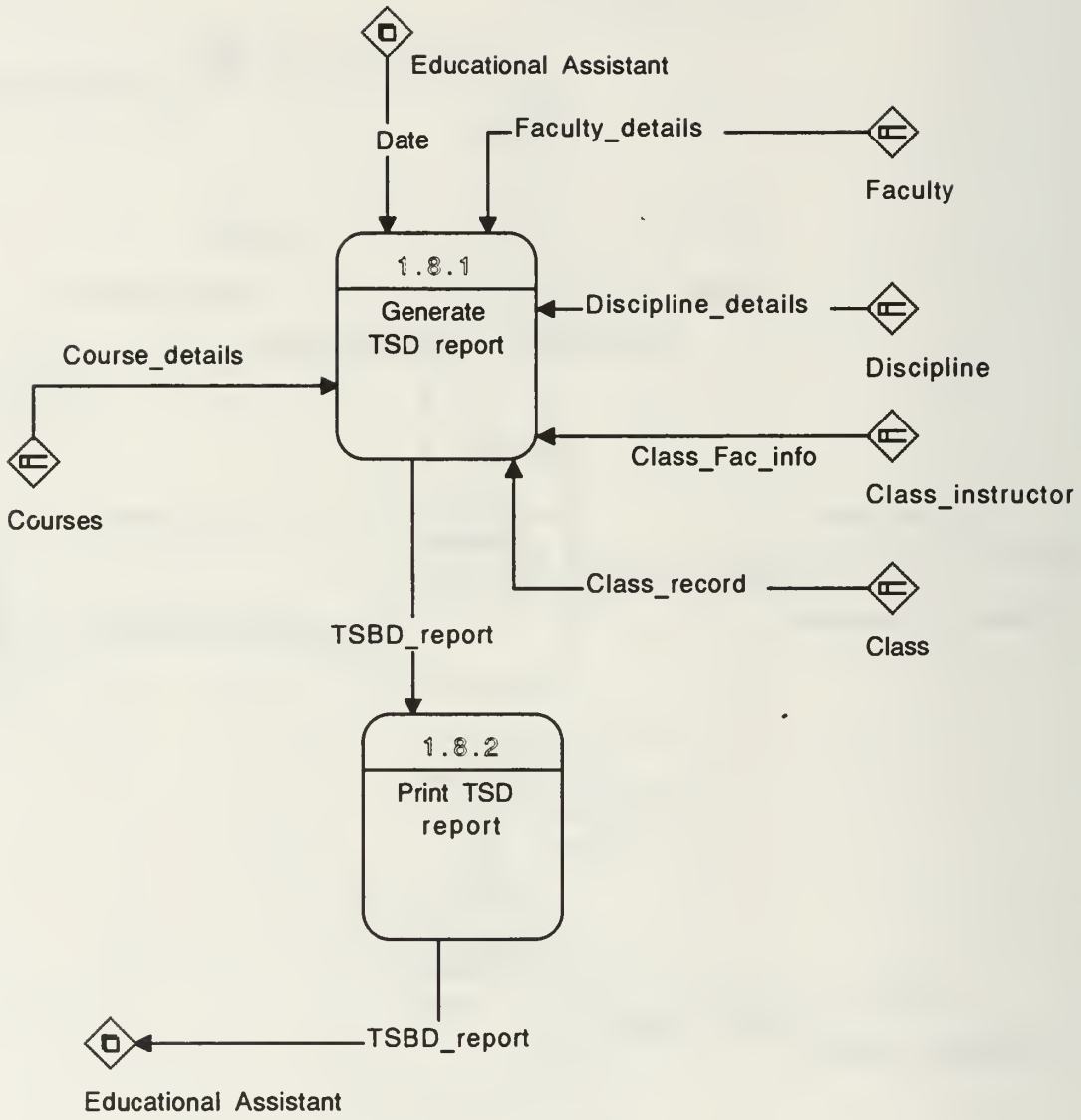

1.3.4
View courses
Course_curric_details
Course_curriculum
course_list
Educational Assistan
Course_curric_details
course_list
1.3.1
Add a course
Course_details
Courses
Course_details
1.3.2
Modify a course
Discipline_details
Discipline
Discipline_details
Discipline
Course_details
Discipline_details
Course_curric_details
1.3.3
Delete a course
Course_curric_details
Course_curriculum
Course_changes
Course_changes
Course_changes
Educational Assistant

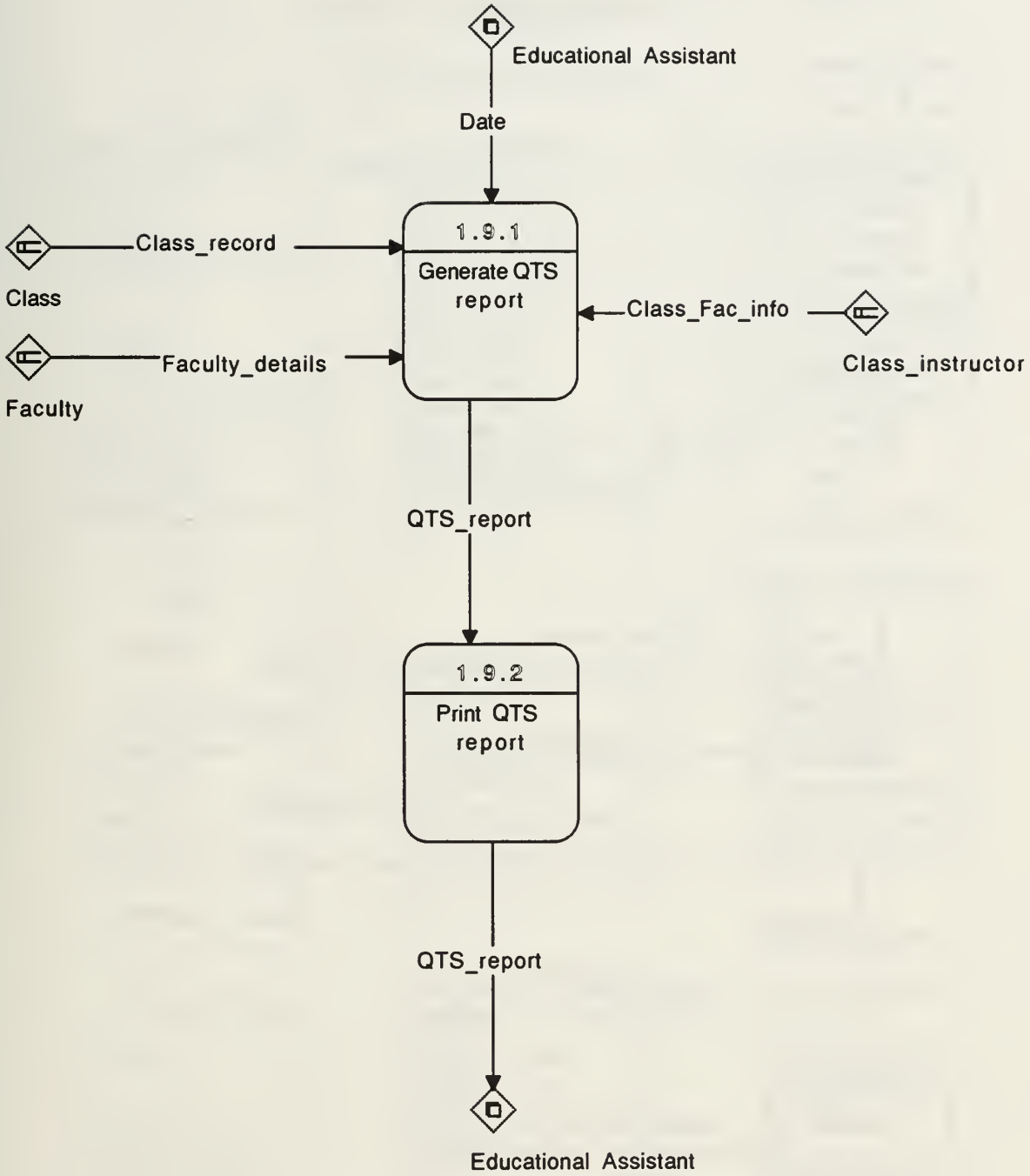

1.4.1
View Disciplines
Discipline_details
1.4.2
Add a discipline
Discipline_details
Discipline_details
1.4.3
Modify a discipline
Discipline
Discipline_details
1.4.4
Delete a discipline
Discipline_details
Discipline_details
Discipline_details
Educational Assistant

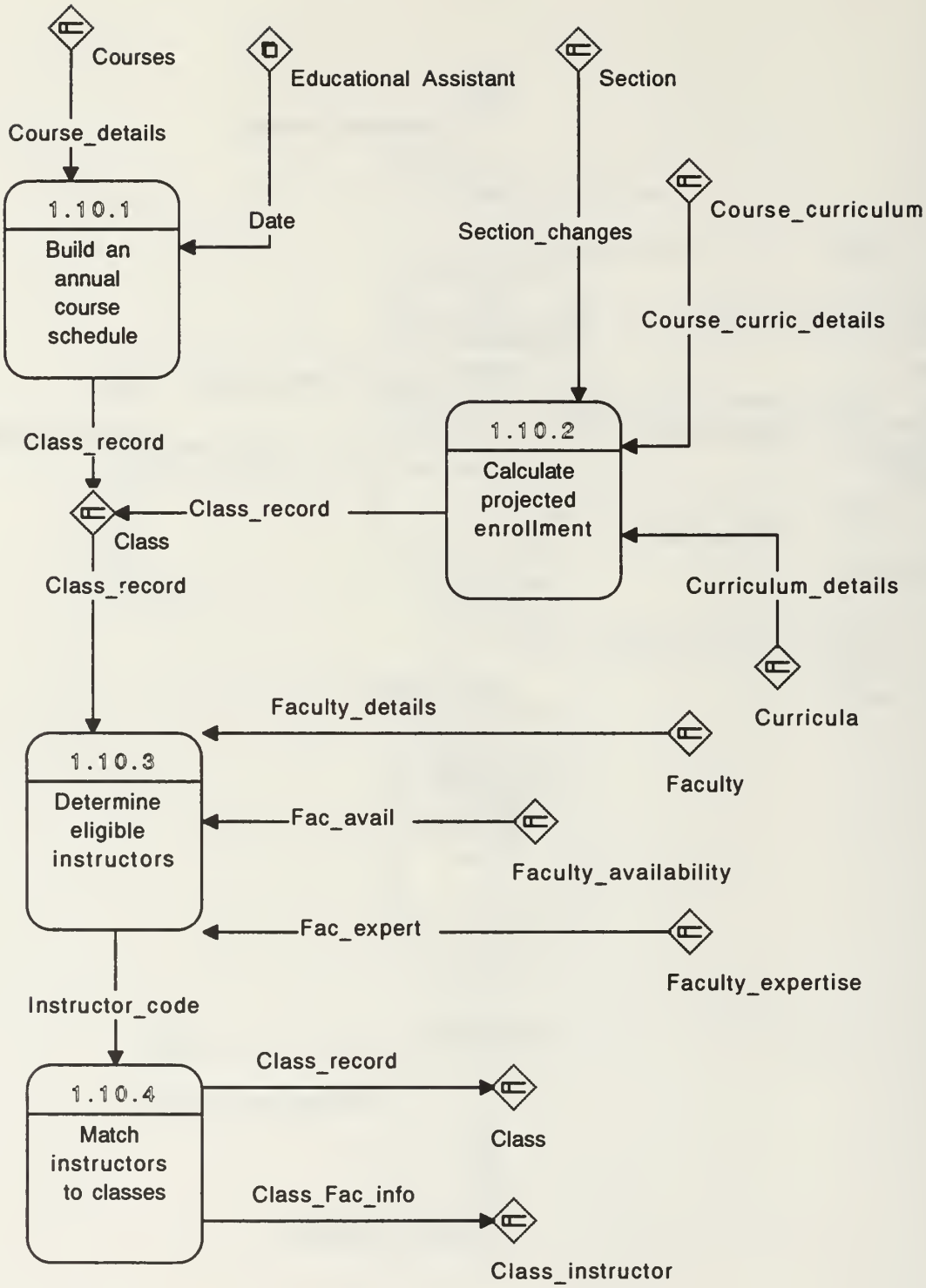

Fac_avail
Faculty_availability
Educational Assistant
Fac_avail
1.5.4
View faculty
Faculty_changes
Fac_expert
Fac_expert
Faculty_expertise
Faculty_details
1.5.1
Add a faculty
Faculty_details
Faculty_details
Faculty
1.5.2
Modify a faculty
Fac_avail
Fac_expert
Faculty_details
Faculty_availability
Fac_avail
Faculty_changes
1.5.3
Delete a faculty
Fac_expert
Faculty_expertise
Faculty_changes
Faculty_changes
Educational Assistant


Educational Assistant
Date
1.6.1
Generate CO report
Class_record
Class
CO_report
1.6.2
Print CO report
CO_report
Educational Assistant


Educational Assistant
Date
Class
Class_instructor
Class_record
Class_Fac_info
1.7.1
Generate
EMY report
m_y_estimate
m_y_estimate
revised_estimate
AS Dept AC/ Instruction
revised_estimate
AS Dept. chairman
EMY_report
1.7.2
Print EMY
report
EMY_report
Educational Assistant

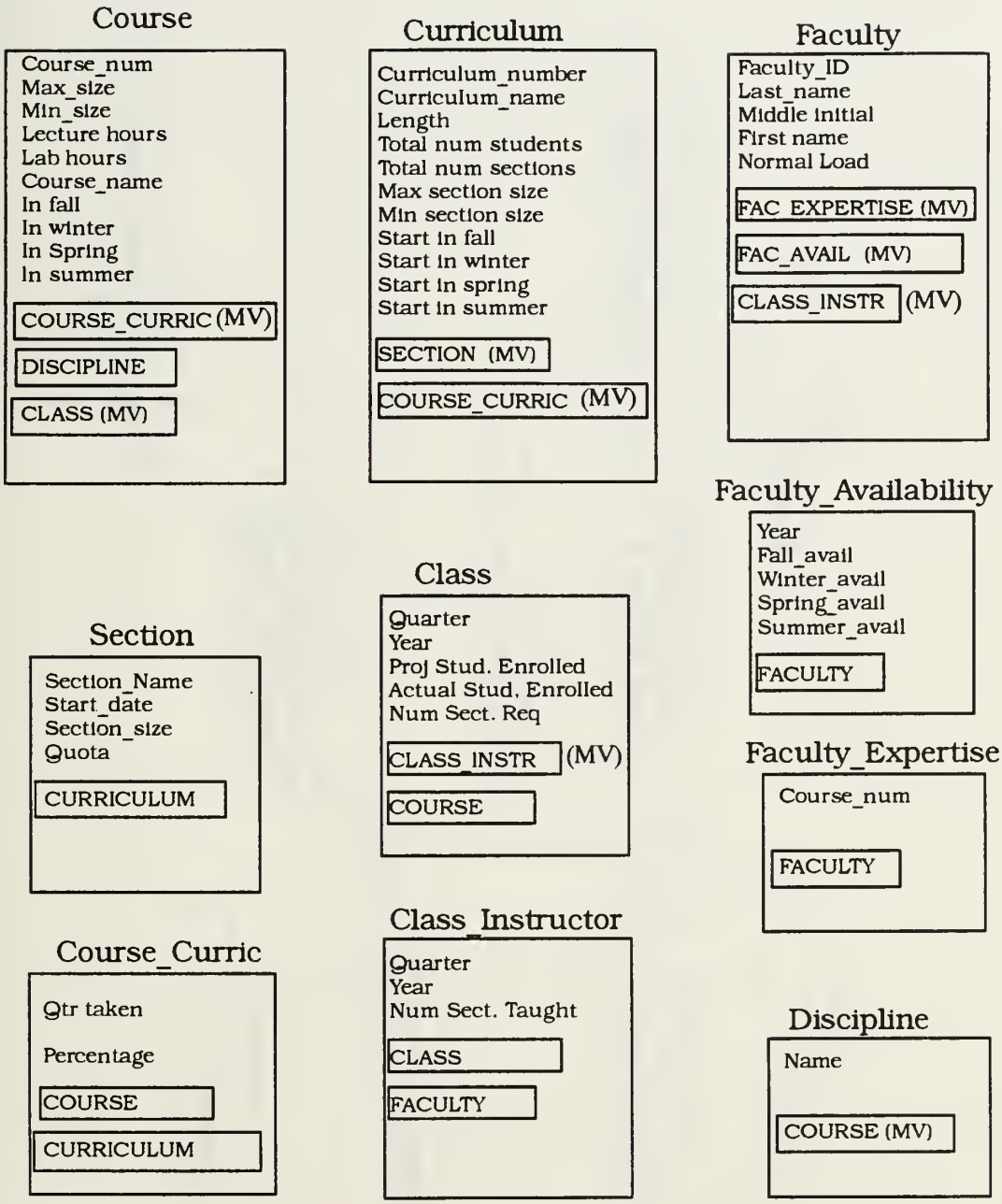

Educational Assistant
Date
Faculty_details
Faculty
1.8.1
Generate
TSD report
Discipline_details
Discipline
Course_details
Courses
Class_Fac_info
Class_instructor
Class_record
Class
TSBD_report
1.8.2
Print TSD
report
TSBD_report
Educational Assistant


Educational Assistant
Date
1.9.1
Generate QTS report
Class_record
Class
Faculty_details
Faculty
Class_Fac_info
Class_instructor
QTS_report
1.9.2
Print QTS report
QTS_report
Educational Assistant


Courses
Educational Assistant
Section
Course_details
1.10.1
Build an annual course schedule
Date
Section_changes
Course_curriculum
Course_curric_details
Class_record
1.10.2
Calculate projected enrollment
Class_record
Class
Class_record
Curriculum_details
Curricula
Faculty_details
Faculty
1.10.3
Determine eligible instructors
Fac_avail
Faculty_availability
Fac_expert
Faculty_expertise
Instructor_code
Class_record
1.10.4
Match instructors to classes
Class
Class_Fac_info
Class_instructor

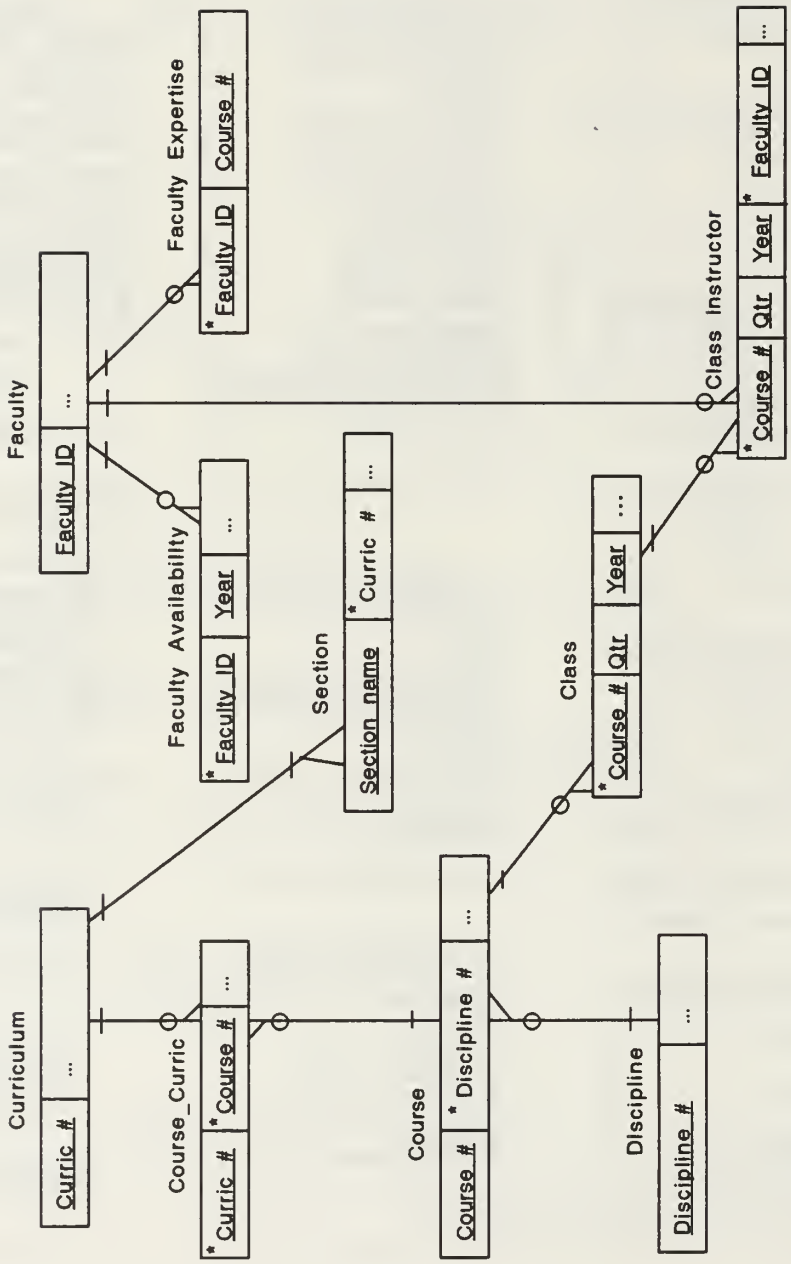
APPENDIX C

OBJECT DIAGRAM



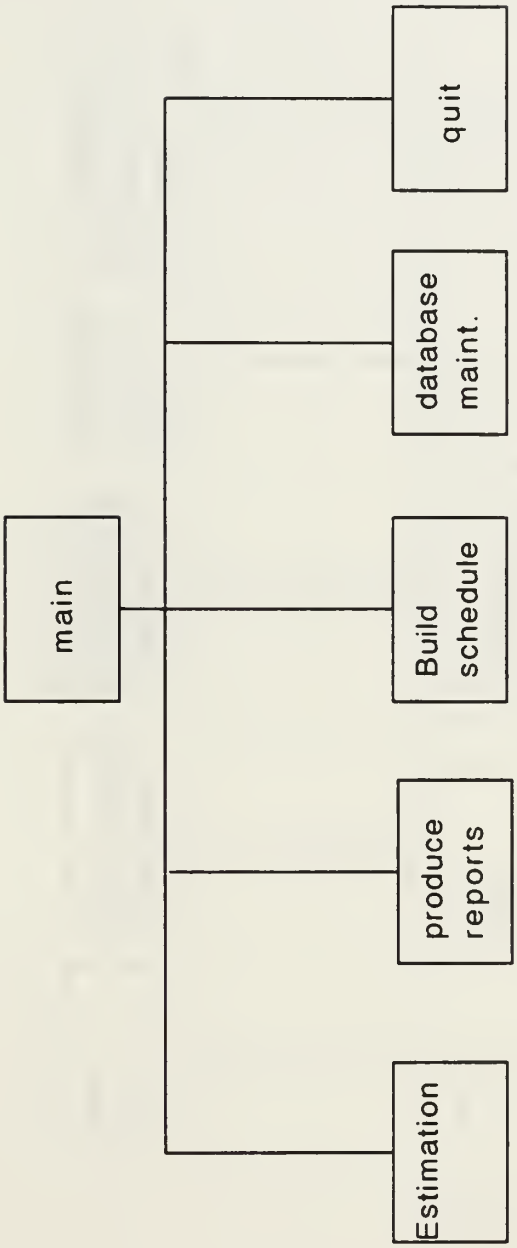
APPENDIX D

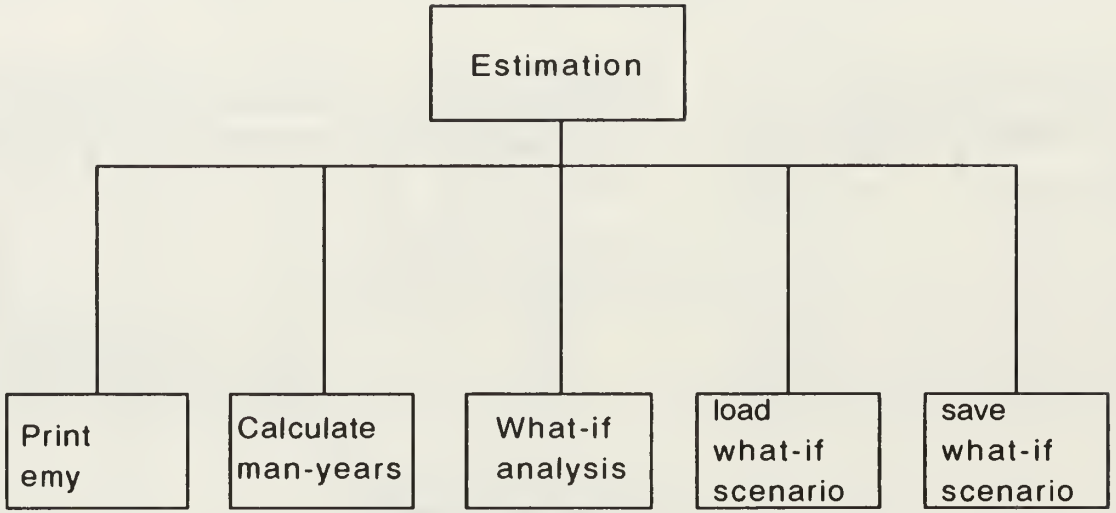
RELATION DIAGRAM

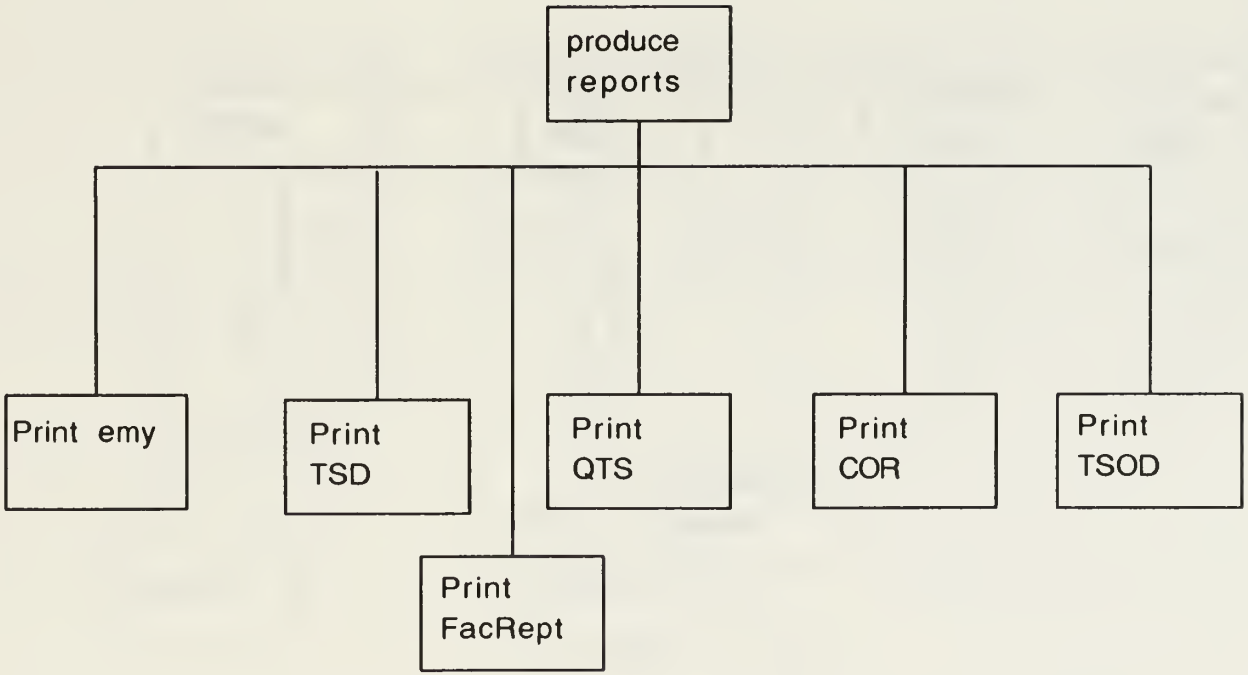


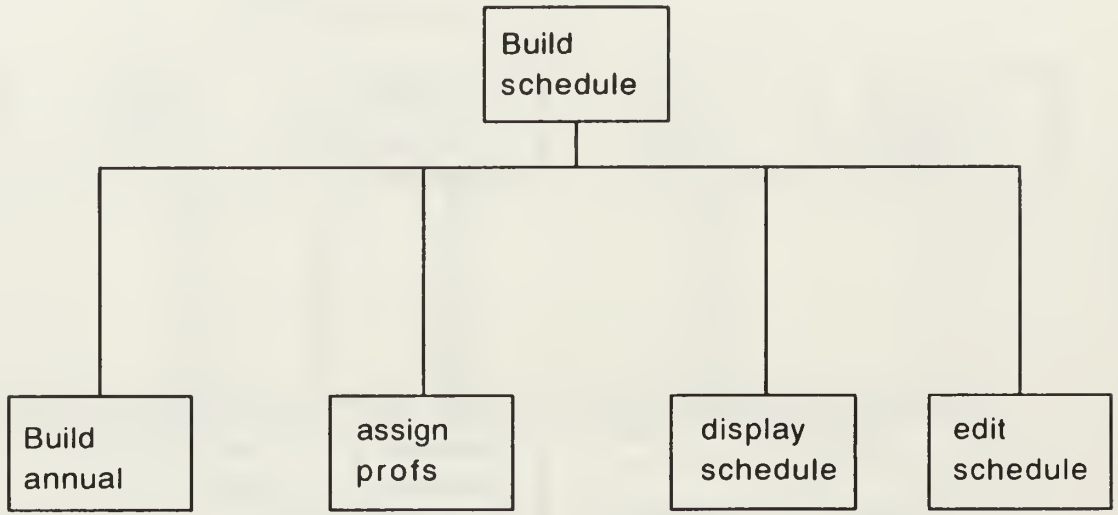
APPENDIX E

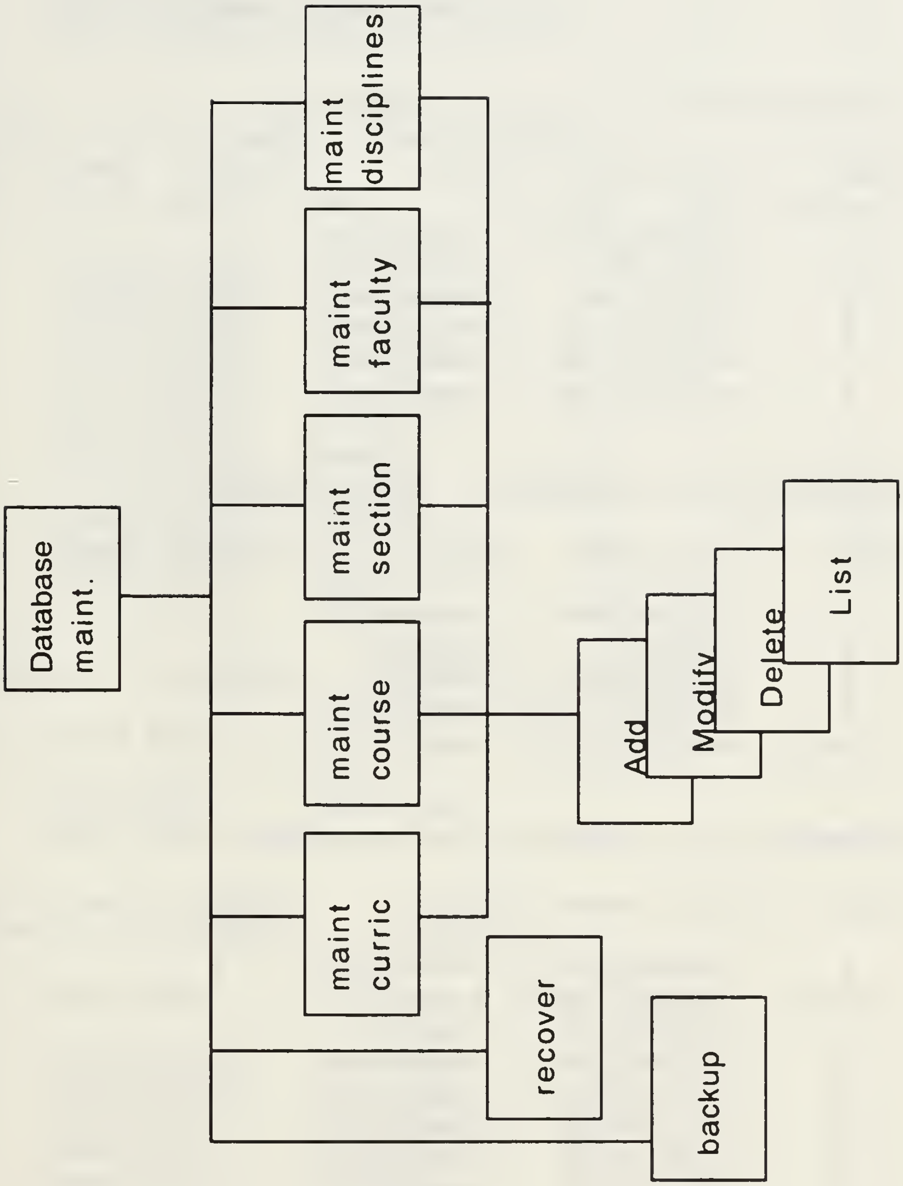
MENU STRUCTURE




Estimation
Print emy
Calculate man-years
What-if analysis
load what-if scenario
save what-if scenario


produce reports
Print emy
Print TSD
Print FacRept
Print QTS
Print COR
Print TSOD


Build schedule
Build annual
assign profs
display schedule
edit schedule


Database maint.
maint curric
maint course
maint section
maint faculty
maint disciplines
recover
backup
Add
Modify
Delete
List

APPENDIX F

DATABASE STRUCTURE

Structure for database: CLASS.DBF

Field	Field Name	Type	Width	Dec	Index
1	COURSE_NUM	Character	6		Y
2	THE_QTR	Numeric	1		Y
3	THE_YEAR	Numeric	4		Y
4	PRJ_STU_EN	Numeric	3		N
5	NUM_SEC_RQ	Numeric	2		N
6	ACT_STU_EN	Numeric	3		N

Structure for database: CLASS.IN.DBF

Field	Field Name	Type	Width	Dec	Index
1	COURSE_NUM	Character	6		Y
2	THE_QTR	Numeric	1		Y
3	THE_YEAR	Numeric	4		Y
4	FACULTY_ID	Character	2		Y
5	NUM_SEC_TA	Numeric	2		N

Structure for database: COURSE.DBF

Field	Field Name	Type	Width	Dec	Index
1	COURSE_NUM	Character	6		Y
2	CRS_MAX_SZ	Numeric	2		N
3	CRS_MIN_SZ	Numeric	1		N
4	COURSE_NAM	Character	35		N
5	CRS_LAB_HR	Numeric	1		N
6	CRS_LEC_HR	Numeric	1		N
7	DISC_NUM	Character	3		Y
8	CRS_IN_FAL	Logical	1		N
9	CRS_IN_WIN	Logical	1		N
10	CRS_IN_SPR	Logical	1		N
11	CRS_IN_SUM	Logical	1		N

Structure for database: CRSE_CUR.DBF

Field	Field Name	Type	Width	Dec	Index
1	COURSE_NUM	Character	6		Y
2	CURRIC_NUM	Character	7		Y
3	QTR_TAKEN	Numeric	1		N
4	CRS_PERCEN	Float	6	4	N

Structure for database: CURRICUL.DBF

Field	Field Name	Type	Width	Dec	Index
1	CURRIC_LEN	Numeric	1		N
2	CURRIC_NAM	Character	25		N
3	CURRIC_NUM	Character	7		Y
4	TOT_NUM_ST	Numeric	4		N
5	TOT_NUM_SE	Numeric	3		N
6	MAX_SEC_SZ	Numeric	2		N
7	MIN_SEC_SZ	Numeric	2		N
8	START_FALL	Logical	1		N
9	START_WIN	Logical	1		N
10	START_SPR	Logical	1		N
11	START_SUM	Logical	1		N

Structure for database: DISCIPLN.DBF

Field	Field Name	Type	Width	Dec	Index
1	DISC_NUM	Character	3		Y
2	DISC_NAME	Character	30		N

Structure for database: FACULTY.DBF

Field	Field Name	Type	Width	Dec	Index
1	FACULTY_ID	Character	2		Y
2	FIRST_NAME	Character	10		N
3	LAST_NAME	Character	15		N
4	INITIAL	Character	8		N
5	NORM_LOAD	Numeric	2		N

Structure for database: FACAVAIL.DBF

Field	Field Name	Type	Width	Dec	Index
1	FACULTY_ID	Character	2		Y
2	YEAR	Numeric	4		Y
3	FALL	Logical	1		N
4	WINTER	Logical	1		N
5	SPRING	Logical	1		N
6	SUMMER	Logical	1		N

Structure for database: FACEXPER.DBF

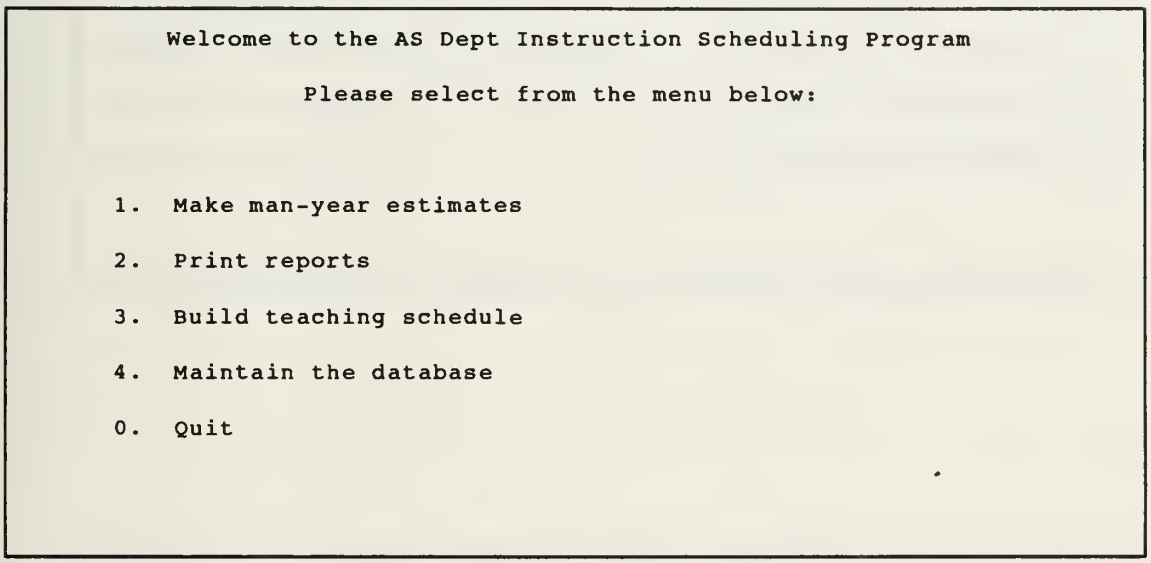
Field	Field Name	Type	Width	Dec	Index
1	FACULTY_ID	Character	2		Y
2	COURS_QUAL	Character	6		Y

Structure for database: SECTION.DBF

Field	Field Name	Type	Width	Dec	Index
1	SEC_NAME	Character	4		Y
2	SEC_SIZE	Numeric	2		N
3	SEC_QUOTA	Numeric	2		N
4	CURRIC_NUM	Character	7		N
5	START_DATE	Date	8		N

APPENDIX G

SCREEN DESIGNS



Estimation Menu

Please select from the menu below:

1. Show current man-years estimate
2. Do What-if analysis
3. Save What-if scenario
4. Load What-if scenario
0. Return to main menu

Produce Reports Menu

Please select from the menu below:

1. Print Estimated Man-Year Report
2. Print Teaching Schedule by Discipline
3. Print Quarterly Teaching Schedule
4. Print Course Offerings Report
5. Print Teaching Schedule for One Discipline
6. Print Faculty Teaching Report
0. Return to main menu

Build an Annual Course Schedule Menu

Please select from the menu below:

1. Build an annual schedule
2. Determine qualified instructors
3. Assign instructors to all classes
4. Assign instructors to one class
0. Return to main menu

Maintain the Database Menu

Please select from the menu below:

1. Maintain CURRICULUM information
2. Maintain COURSE information
3. Maintain SECTION information
4. Maintain FACULTY information
5. Maintain DISCIPLINE information
6. Backup the database (to floppy)
7. Restore the database (from floppy)
0. Return to main menu

Estimated Man-Years for Academic Year 199X

Comments: Put your comments here

	Discipline:	Fall	Winter	Spring	Summer	Total
367	Information Systems	99.9	99.9	99.9	99.9	99.9
400	Management Policy	88.8	88.8	88.8	88.8	88.8

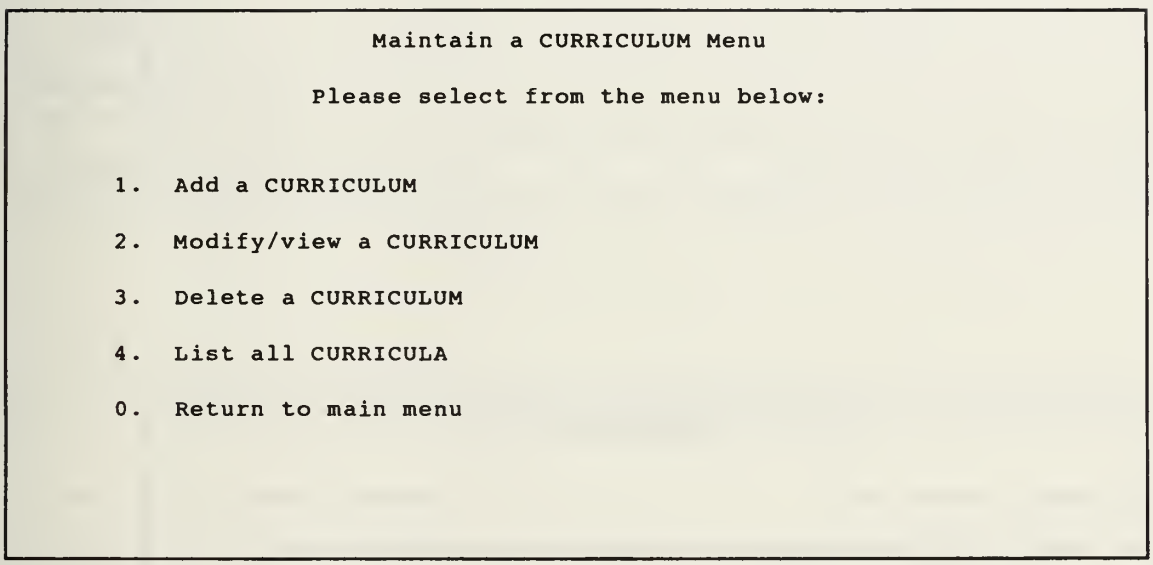
Subtotal	99.9	99.9	99.9	99.9	
Grand Total	99.9				

Enter incoming students

Quarter

Curriculum	Next	N+1	N+2	N+3	N+4	N+5	N+6	N+7
367	99	99	99	99	99	99	99	99
620	99	99	99	99	99	99	99	99
817	99	99	99	99	99	99	99	99
837	99	99	99	99	99	99	99	99

Press F1 to run WHAT-IF EMY Press F2 to run WHAT-IF TSD
Press F3 to save scenario Press F4 to load scenario


Maintain a CURRICULUM Menu
Please select from the menu below:
1. Add a CURRICULUM
2. Modify/view a CURRICULUM
3. Delete a CURRICULUM
4. List all CURRICULA
0. Return to main menu

```
                    Add a CURRICULUM

Enter curriculum number: 999

Curriculum name: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Curriculum length : 9

Does curriculum start in:
  Fall   : Y
  Winter: N
  Spring: N
  Summer: Y
```

```
                    Delete a CURRICULUM

Enter curriculum number: 999

Curriculum name: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

WARNING: Are you sure: N
```

```
                              Add a COURSE


Enter course number: AA9999
Course name: XXXXXXXXXXXXXXXXXXXXXXXXXX

                                          Curriculum  Required    Qtr
                                            367          Y         3
Max size: 99                                620          N
Min size:  9                                817USMC      Y         1
                                            -
When is course offered:
  Fall   : Y
  Winter: N
  Spring: N
  Summer: Y

Lecture hours: 9
Lab hours    : 9

Discipline: 999
```

```
                             Delete a COURSE

Enter course number: AA9999

Course name: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

WARNING: Are you sure: N
```

```
                        Add a SECTION

Enter section name: AA99

Section size:       99

Section quota:      99

Curriculum number:  999
```

```
                        Delete a SECTION

Enter the name of the section you want to delete: AA99

WARNING: Are you sure: N
```

```
                    Add a DISCIPLINE

Enter discipline number:  999

Enter discipline name: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
                    Delete a DISCIPLINE

Enter the discipline number: 999

WARNING: Are you sure: N
```

```
                        Add a FACULTY

Enter faculty code: AA

                                                    Expertise:
First name: AAAAAAAAAA                            Course# AA9999
Middle initial/name: AAAAAAA                      Course# AA9999
Last name:  AAAAAAAAAA                            Course# AA9999
                                                  Course# AA9999
Normal course load: 9

                                                Is this faculty member
                                               available during AY199X

                                                  Fall      N
                                                  Winter    N
                                                  Spring    N
                                                  Summer    N
```

```
                        Delete a FACULTY

Enter the faculty code of the faculty member you want to delete: AA

WARNING: Are you sure: N
```

APPENDIX H

SAMPLE REPORTS

DEPARTMENT OF ADMINISTRATIVE SCIENCES
TEACHING SCHEDULE BY DISCIPLINES
AY 1991-1992

	FALL	WINTER	SPRING	SUMMER
367 Information Systems				
IS0123 (0-0)		Lindsay(6)		Lindsay(12)
IS2000 (3-1)	Knight(2)		Knight(2)	
IS3000 (4-0)				
IS3020 (3-2)		Zweig(2) Sengupta(1)		Zweig(2)
400 Management Policy				
MN4105 (4-0)	Roberts, N.(2) Elster(2) Evered(1)		Roberts, N.(2) Evered(2)	
410 Organization Behavior				
MN3105 (4-0)	Barrett(2) Hocevar(2) Thomas, K.(2)	Crawford(2)	Hocevar(1) Thomas, K.(2)	Barrett(2)
MN4125 (4-0)	Evered(1)			Evered(1)
600 Communications				
AS1701 (4-0)	Dresser(2)		Dresser(2)	Dresser(1)

Estimated Man-Years for Academic Year 199X

Comments: Put your comments here

	Discipline:	Fall	Winter	Spring	Summer	Total
367	Information Systems	99.9	99.9	99.9	99.9	99.9
400	Management Policy	88.8	88.8	88.8	88.8	88.8
-						
	Subtotal	99.9	99.9	99.9	99.9	
	Grand Total	99.9				

NAVAL POSTGRADUATE SCHOOL
Monterey, CA 93943-5000

NC4 AS/DK
3 March 90

MEMORANDUM

From: Chairman, Department of Administrative Sciences
To: Curricular Officers/Academic Associates
Department Chairman

Subject: COURSE OFFERINGS, Spring Quarter, 1990

1. The following courses will be offered by the Department of Administrative Sciences for the Spring quarter (beginning Mar 1990):

IS2000	IS4183	MN3001
IS3000	IS4320	MN4154

2. The following courses will not be offered by the administrative Sciences Department for the Spring quarter:

IS3170	IS4300	MN2150
IS4200		

Daniel R. Dolk
Associate Chair for Instruction
Administrative Sciences Department

Dist: codes 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, CS, MA, AS/BO, AS/EB, AS/EV, AS/FM, AS/LT, AS/MG, AS/SA, QR, NS, PH, EC, MR, AA, OC, ME, AW, SP, EW, CC, 144A

NAVAL POSTGRADUATE SCHOOL
Monterey, CA 93943-5000

NC4 AS/DK
3 May 91

MEMORANDUM

From: Chairman, Department of Administrative Sciences
To: Curricular Officers/Academic Associates
Department Chairman

Subject: COURSE OFFERINGS, Summer Quarter, 1991

1. The following courses will be offered by the Department of Administrative Sciences for the Spring quarter (beginning Mar 1990):

IS2000	IS4200	MN3001
IS3000	IS4320	MN4154

2. The following courses will not be offered by the administrative Sciences Department for the Spring quarter:

IS3170	IS4300	MN2150
IS4183		

Daniel R. Dolk
Associate Chair for Instruction
Administrative Sciences Department

Dist: codes 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, CS, MA, AS/BO, AS/EB, AS/EV, AS/FM, AS/LT, AS/MG, AS/SA, QR, NS, PH, EC, MR, AA, OC, ME, AW, SP, EW, CC, 144A

DEPARTMENT OF ADMINISTRATIVE SCIENCES
TEACHING SCHEDULE BY DISCIPLINES
AY 1991-1992

FALL	WINTER	SPRING	SUMMER
367 Information Systems			

IS0123 (0-0)			
IS2000 (3-1)			
IS3000 (4-0)			
IS3020 (3-2)			
IS3100 (3-0)			
IS3170 (4-0)			
IS3183 (4-0)			
400 Management Policy			

MN4105 (4-0)			
410 Organization Behavior			

MN3105 (4-0)			
MN4125 (4-0)			
600 Communications			

AS1701 (4-0)			

DEPARTMENT OF ADMINISTRATIVE SCIENCES
FACULTY TEACHING REPORT
AY 1989-1990

	FALL	WINTER	SPRING	SUMMER
Abdel-Hamid			IS3183(2)	
	IS4300(2)			
Bhargava			IS4183(2)	
Boger		MN4373(1)		
Bui		IS3000(1)		
		IS4185(1)		IS4185(2)
Carrick				MN2031(1)
	MN4145(2)		MN4145(2)	
Crawford	MN3105(2)			
		MN2113(1)		MN2113(1)
		MN4119(1)		MN4119(1)
Dolk			IS2000(2)	
			IS4320(1)	
Dresser	AS1501(1)	AS1501(1)		
			AS1701(2)	AS1701(1)
Eberling	MN4154(4)		MN4154(3)	
Eitelberg		MN2111(1)		MN2111(1)
	MN2112(1)		MN2112(1)	
		MN4106(1)		MN4106(1)
Elster	MN4105(2)		MN4105(2)	
Evered	MN4105(2)		MN4105(1)	
		MN3333(1)		
		MN4155(1)		
Fann		MN3333(2)		MN3333(2)

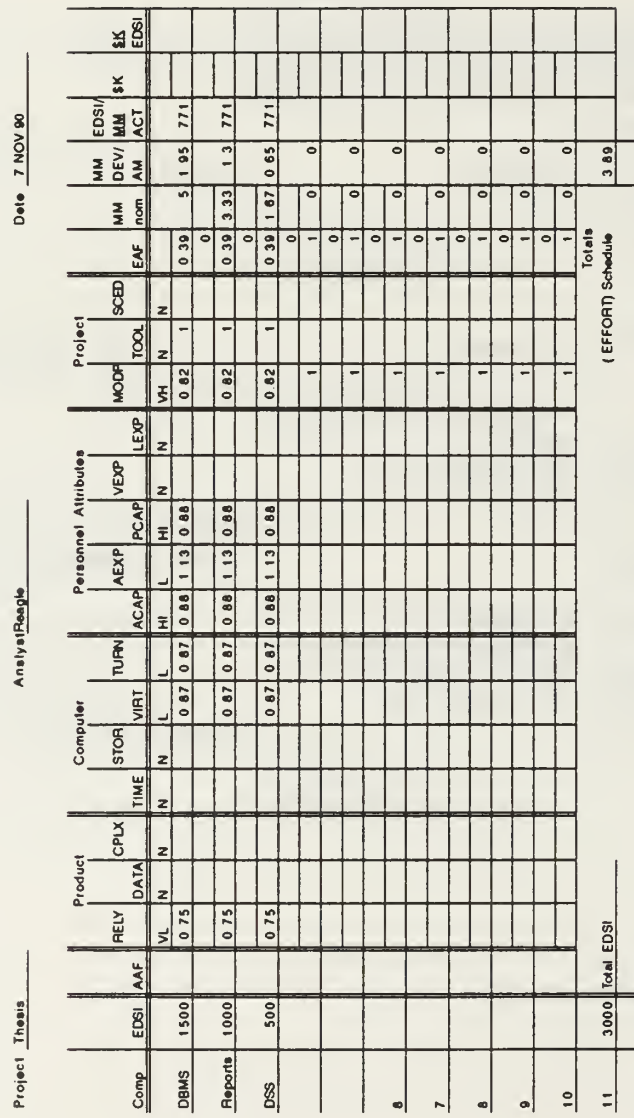
DEPARTMENT OF ADMINISTRATIVE SCIENCES
TEACHING SCHEDULE FOR
DEPT: 367 Information Systems

	FALL	WINTER	SPRING	SUMMER
AY 1989-1990				
367 Information Systems				
IS0123 (0-2)		Lindsay(6)		Lindsay(10)
IS2000 (3-0)	Knight(2)		Dolk(2)	
IS2100 (0-2)		Sahlman(2)		Sahlman(2)
IS3000 (4-0)		Bui(1)		Knight(1)
IS3020 (3-2)		Sengupta(2)		Sengupta(2)
IS3170 (4-0)		Haga(2)		Haga(2)
IS3183 (4-0)	Haga(2)		Haga(1)	
	Kamel(1)		Abdel-Hamid(2)	
	Frew(1)			
IS3220 (3-2)		Knight(1)		
IS3502 (4-0)		Schneidewind(1)		Suh(3)
IS3503 (3-2)		Schneidewind(1)		
IS4182 (4-0)		Zviran(2)		Zviran(1)
IS4183 (4-0)	Kamel(1)		Bhargava(3)	

APPENDIX I

COCOMO

A. Estimate 1



Project Thesis Analyst Reagle Date 7 NOV 90

Comp	EDSI	AAF	Product RELY	Product DATA	Product CPLX	Computer TIME	Computer STOR	Computer VIRT	TURN	Personnel Attributes ACAP	Personnel Attributes AEXP	Personnel Attributes PCAP	Personnel Attributes VEXP	Personnel Attributes LEXP	Project MODP	Project TOOL	Project SCED	EAF	MM nom	MM DEV/ MM AM	EDSI/ MM ACT	$K	$K EDSI
			VL	N	N	N	N	L	L	HI	L	HI	N	N	VH	N	N						
DBMS	1500		0 75					0 87	0 87	0 88	1 13	0 88			0 82	1		0 39	5	1 95	771		
																		0					
Reports	1000		0 75					0 87	0 87	0 88	1 13	0 88			0 82	1		0 39	3 33	1 3	771		
																		0					
DSS	500		0 75					0 87	0 87	0 88	1 13	0 88			0 82	1		0 39	1 67	0 65	771		
																		0					
															1			1	0	0			
																		0					
															1			1	0	0			
																		0					
6															1			1	0	0			
																		0					
7															1			1	0	0			
																		0					
8															1			1	0	0			
																		0					
9															1			1	0	0			
																		0					
10															1			1	0	0			
																		Totals					
11	3000	Total EDSI																(EFFORT) Schedule		3 89			
12	10	(MM)nom						Organic				Development Mode Organic						TDEV (months)		4 19			
13	300	(EDSI/MM)nom (Productivity)						Semidetached															
								Embedded															

MM Coefficient 3 20
MM Exponent 1 05

Project: Thesis Analyst: Reagle Date: 7 Mar 91

			Product			Computer				Personnel Attributes					Project							
Comp	EDSI	AAF	RELY	DATA	CPLX	TIME	STOR	VIRT	TURN	ACAP	AEXP	PCAP	VEXP	LEXP	MODP	TOOL	SCED	EAF	MM nom	MM DEV/ MM	EDSI/ MM ACT	$K	$K EDSI
			VL	N	N	N	N	L	L	HI	L	HI	N	N	VH	N	N						
DBMS	1500		0.75					0.87	0.87	0.86	1.13	0.86			0.82	1		0.39	5	1.95	771		
																		0					
Reports	1000		0.75					0.87	0.87	0.86	1.13	0.86			0.82	1		0.39	3.33	1.3	771		
																		0					
Estimation	1000		0.75					0.87	0.87	0.86	1.13	0.86			0.82	1		0.39	3.33	1.3	771		
																		0					
Schedule	1000		0.75					0.87	0.87	0.86	1.13	0.86			0.82	1		0.39	3.33	1.3	771		
																		0					
															1			1	0	0			
																		0					
6															1			1	0	0			
																		0					
7															1			1	0	0			
																		0					
8															1			1	0	0			
																		0					
9															1			1	0	0			
																		0					
10															1			1	0	0			
11	4500	Total EDSI																Totals (EFFORT) Schedule		5.84			
12	15	(MM)nom																TDEV (months)		4.89			
13	300	(EDSI/MM)nom (Productivity)																					

Organic
Semidetached
Embedded

Development Mode: Organic

MM Coefficient: 3.20
MM Exponent: 1.05

APPENDIX J

GROWTH ESTIMATE

Size of Database

	Data Store	Record size	# records	# fields	Header size	Total	Source
1	Course	53	110	11	386	6216	NPS Course Catalog
2	Course_curriculum	20	4500	4	162	90162	NPS Course Catalog
3	Curriculum	48	18	11	386	1250	NPS Course Catalog
4	Discipline	33	10	2	98	428	TSD
5	Faculty	36	81	5	194	3110	NPS Course Catalog
6	Faculty_availability	8	648	6	226	5410	8 qtrs * 81 profs
7	Faculty_expertise	8	405	2	98	3338	5 expertises / prof
8	Section	23	36	5	194	1022	NPS Course Catalog
9	Class	18	484	6	226	8938	TSD
10	Class_instructor	14	484	5	194	6970	TSD
						126,844	

Growth of database

ACT faculty: 30% per year	3557
ACT Section: 16 new sections/year	16352
ACT Class: 100% per year	8938
ACT Class_instructor : 100% per year	6970
Annual growth	**28%**

APPENDIX K

LIST OF PROGRAM MODULES

	Module	dBASE Name
	main	main
Cat 1	db_maint	db_maint
	maint_course	maint_co
	add_course	add_cour
	mod_course	mod_cour
	del_course	del_cour
	list_course	list_cou
	maint_curric	maint_cu
	add_curric	add_curr
	mod_curric	mod_curr
	del_curric	del_curr
	list_curric	list_cur
	maint_faculty	maint_fa
	add_faculty	add_facu
	mod_faculty	mod_facu
	del_faculty	del_facu
	list_faculty	list_fac
	maint_section	maint_se
	add_section	add_sect
	mod_section	mod_sect
	del_section	del_sect
	list_section	list_sec
	maint_disc	maint_di
	add_disc	add_disc
	mod_disc	mod_disc
	del_disc	del_disc
	list_disc	list_dis
	backup DB	backup
	restore DB	recover
	----------	
	sec2curr	sec2curr
	message	message
	chooser	chooser
	error	error
	find quarter	fquarter
	check name	chkname
	get_id	get_id
	check_row	chk_row
	validate disc	val_disc
	get last name	get_last
	disc name	dis_name

	Module	dBASE Name
Cat 2	reports_menu	prod_rep
	print_cor	prt_cor
	calculate offered	calc_neq
	calc. not offered	calc_eq
	cor_hdr	cor_hdr
	cor_ftr	cor_ftr
	print_qts	prt_qts
	print_tsbd	prt_tsbd
	tsod_driver	tsod_drv
	ts_one_disc	one_disc
	all_faculty_report	fac_rept
Cat 3	estimate menu	est_menu
	print_emy	prt_emy
	calc_emy	calc_emy
	tsbd_if	tsbd_if
	what_if	what_if
	save scenario	save_if
	load scenario	load_if
Cat 4	build schedule	build_sk
	build_annual	build_an
	build_it	build_it
	calc_proj_enroll	calc_enr
	assign_all	ass_all
	get_info	get_info
	assign_one	ass_one
	add a class	add_clas
	edit a class	edit_cla
	delete a class	del_clas
quit		quit

APPENDIX L

USER MANUAL

System Overview

STARTING THE PROGRAM

To start the program, type "DO MAIN" from the dot prompt. This will start the ISP 2.0 program running. Wait a few seconds while the program loads into main memory and opens database files.

Data entry conventions

Some data fields have default entries. To accept those values, use the TAB or ENTER keys. If you want to put a different value in, type it in. Some fields have variable length information (Last name, for example). After entering information in those fields, you must press ENTER to continue. When a fixed length field is filled, you automatically move to the next field.

Navigation conventions

To move through the menus, press the number key associated with the action you want. Press '0' or ESCAPE to move up one level in the hierarchy. You can use the TAB and arrow keys to move through a data entry screen.

MAIN MENU

The program can be thought of as four distinct applications: man-year estimation, scheduling, report generation, and database management. The menu uses a hierarchical structure (See Appendix A). At the top level, you can select any of the four applications to work on. The first option concerns the estimation of faculty man-years. The second area concerns printing of the various reports. The third area covers the establishment and maintenance of a teaching schedule. The fourth area allows you to maintain a current database by adding new faculty, deleting old courses, deleting student sections that have graduated,etc. The main menu appears below.

```
        Welcome to the AS Dept Instruction Scheduling Program

                Please select from the menu below:


    1.  Make man-year estimates

    2.  Print reports

    3.  Build teaching schedule

    4.  Maintain the database

    0.  Quit
```

SUBORDINATE MENUS

The following four menus appear one level below the main menu. To reach one, select a main menu option (by pressing 1-4). Each menu corresponds to an application.

Man-year estimation

```
                    Estimation Menu

              Please select from the menu below:


    1.  Show current man-years estimate

    2.  Do What-if analysis

    3.  Save What-if scenario

    4.  Load What-if scenario

    5.  Generate TSD from What-if scenario

    0.  Return to main menu
```

Print reports

```
                    Produce Reports Menu

              Please select from the menu below:


    1.  Print Estimated Man-Year Report

    2.  Print Teaching Schedule by Discipline

    3.  Print Quarterly Teaching Schedule

    4.  Print Course Offerings Report

    5.  Print Teaching Schedule for One Discipline

    6.  Print Faculty Teaching Report

    0.  Return to main menu
```

Build an annual schedule

```
                Build an Annual Course Schedule Menu

                Please select from the menu below:


      1.  Build an annual schedule
      2.  Determine qualified instructors
      3.  Add instructor to class
      4.  Change instructor data
      5.  Delete instructor from class
      6.  Add class to schedule
      7.  Change class data
      8.  Delete class from schedule
      0.  Return to main menu
```

Maintain the database

```
                Maintain the Database Menu

                Please select from the menu below:


    1.  Maintain CURRICULUM information

    2.  Maintain COURSE information

    3.  Maintain SECTION information

    4.  Maintain FACULTY information

    5.  Maintain DISCIPLINE information

    6.  Backup the database (to floppy)

    7.  Restore the database (from floppy)

    0.  Return to main menu
```

Backup the database

The program allows you to make backups of the database to floppy disk. Select BACKUP THE DATABASE from the MAINTAIN THE DATABASE menu. You will be prompted to insert a formated disk into the "A" drive. We recommend you backup the database at least once a week. Use two or three disks, and rotate your backups between those disks. If the database becomes corrupted or lost, and recovery is necessary, follow the procedure under "Restore the Database."

Restore the database

To restore the database to the last saved state, insert the most recent backup floppy disk into the "A" disk drive, and select RESTORE THE DATABASE from the MAINTAIN THE DATABASE menu. All databases and index files will be copied from the floppy disk to the hard disk. Any changes made to the original database since the last backup will be lost.

Quitting the program

To quit the program, select "Quit," from the main menu and the program will close all open databases, and clean up where necessary.

A detailed discussion of each application is presented in the following sections.

MANAGING THE DATABASE

This application is reached through menu option 4 of the main menu. The functions *Add*, *Modify*, and *Delete* can be performed on the following databases: *Section*, *Faculty*, *Course*, *Curriculum*, *Discipline*. An example of such a menu is displayed below.

```
                Maintain a CURRICULUM Menu

              Please select from the menu below:

     1.  Add a CURRICULUM

     2.  Modify/view a CURRICULUM

     3.  Delete a CURRICULUM

     4.  List all CURRICULA

     0.  Return to main menu
```

MAINTAIN STUDENT SECTIONS

The numbers of students is a critical factor to the success of the program's model. Every quarter you must keep track of the number of students added to a curriculum. The curricular officers will provide you with this information. To ensure optimal performance, all sections expected within the upcoming academic year must be entered.

Add

You will be adding student sections every quarter. Some curricula start annually, some semi-annually. Begin by entering the section name. This is a four character code consisting of two letters and two numbers (e.g., PL01). The system will check to ensure that the section name does not exist. After the section name is validated,

enter the number of students who actually start in the section, the section quota, and the date the section starts. The curriculum is automatically computed based on the section name, and should not be changed. The *Add Section* screen is displayed below.

```
                          Add a SECTION

Enter section name: AA99

Section size:       99

Section quota:      99

Curriculum number:  999
```

Modify

This screen is identical in format to the *Add Section* screen. Begin by entering the section name. The system will check to ensure the section exists. After the section name is validated, you can view or edit the size, quota, or starting date.

Delete

You should delete sections when the students graduate. Enter the section name. The system will check to ensure the course exists. If the section does exist, a message will appear asking you if you are sure you want to delete the section. If the section does not exist, an error message will be displayed. The *Delete Section* screen is displayed below.

```
                    Delete a SECTION

Enter the name of the section you want to delete: AA99

WARNING: Are you sure: N
```

List

You have the option of displaying a list of all sections to the screen, to the printer, or sending it a text file on disk. Printing the list prevents further use of your computer until printing concludes (unless you have a buffer or print spooler).

MAINTAIN FACULTY

A faculty member is identified by a unique faculty code. Personal data about the faculty, which includes name, course load, expertise (courses he/she can teach), and availability, are maintained by this application. If you do not maintain accurate faculty availability information, this program will give erroneous results.

Add

Add a faculty member upon their arrival at the institution. Enter the faculty code. The system will check to ensure that the code does not exist. After the faculty code is validated, enter their name, course load, expertise and availability. Expertise and

availability fields will not appear until after the course load has been entered. The *Add Faculty* screen is displayed below.

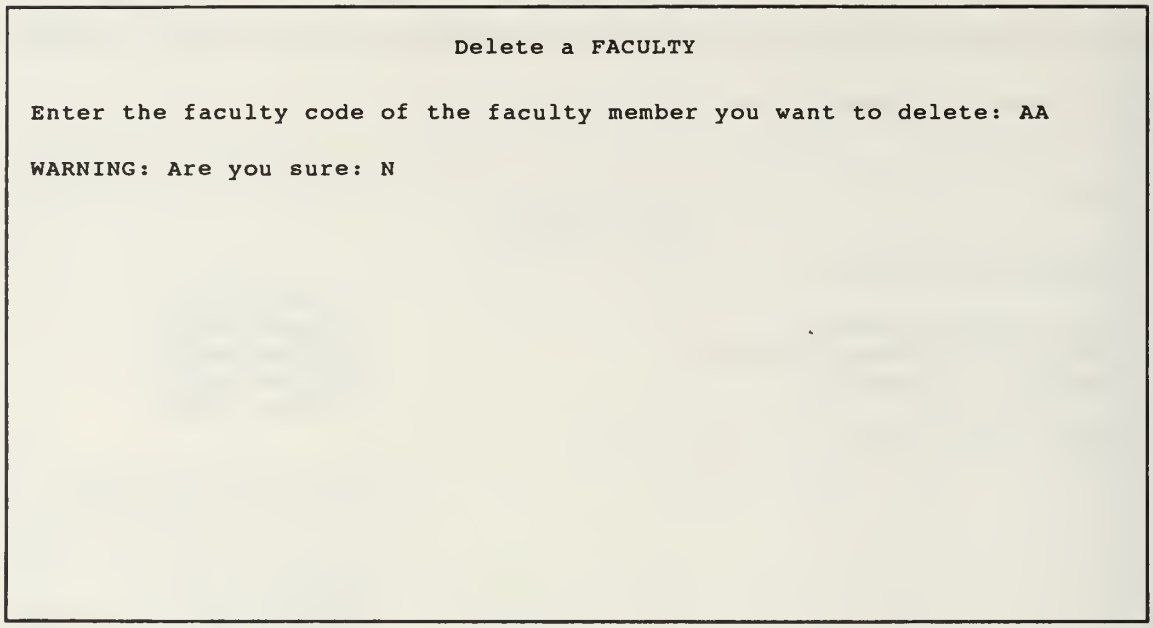
```
                              Add a FACULTY

Enter faculty code: AA

                                                      Expertise:
First name: AAAAAAAAAA                              Course# AA9999
Middle initial/name: AAAAAAA                        Course# AA9999
Last name:  AAAAAAAAAA                              Course# AA9999
                                                    Course# AA9999
Normal course load: 9

                                                 Is this faculty member
                                                available during AY199X

                                                    Fall      N
                                                    Winter    N
                                                    Spring    N
                                                    Summer    N
```

Modify

This screen is identical in format to the *Add Faculty* screen. Enter the faculty code. The system will check to ensure that the faculty code does exist. After the faculty code is validated, you may view or edit any of the fields except faculty code.

Delete

Delete faculty when they depart permanently. For faculty who go on sabbatical or visit other institutions, modify their availability. To delete a faculty, enter the faculty code. The system checks to ensure the faculty exists, and displays the faculty name. A message will appear asking you if you are sure you want to delete. The *Delete Faculty* screen is displayed below.



List

You have the option of displaying a list of all faculty to the screen, to the printer, or sending it a text file on disk. Printing the list prevents further use of your computer until printing concludes (unless you have a buffer or print spooler).

MAINTAIN COURSES

Courses in this file refer to the entity as described in the NPS course catalog. Courses are offered many times per year. We distinguish a course from a class in that a class is an *instance* of a course. For example, IS2000 is a course. IS2000 taught in Fall 1990 is a class.

Add

Enter the course number. The system will check to ensure that the number does not exist. After the course number is validated, enter the course name, and the maximum and minimum number of students permitted in the course. Default values are provided. Enter the quarters in which the course is offered by typing 'Y' if offered, and 'N' if not offered. Enter the lecture and lab hours; again, default values are provided. Enter the discipline number for which this course is associated. If the course is required for a

curriculum, enter a 'Y.' You will then be required to enter the quarter (from the curriculum matrix) it is required in. If the course is not required, enter 'N' and you will be prompted for the next curriculum. The *Add Course* screen is displayed below.

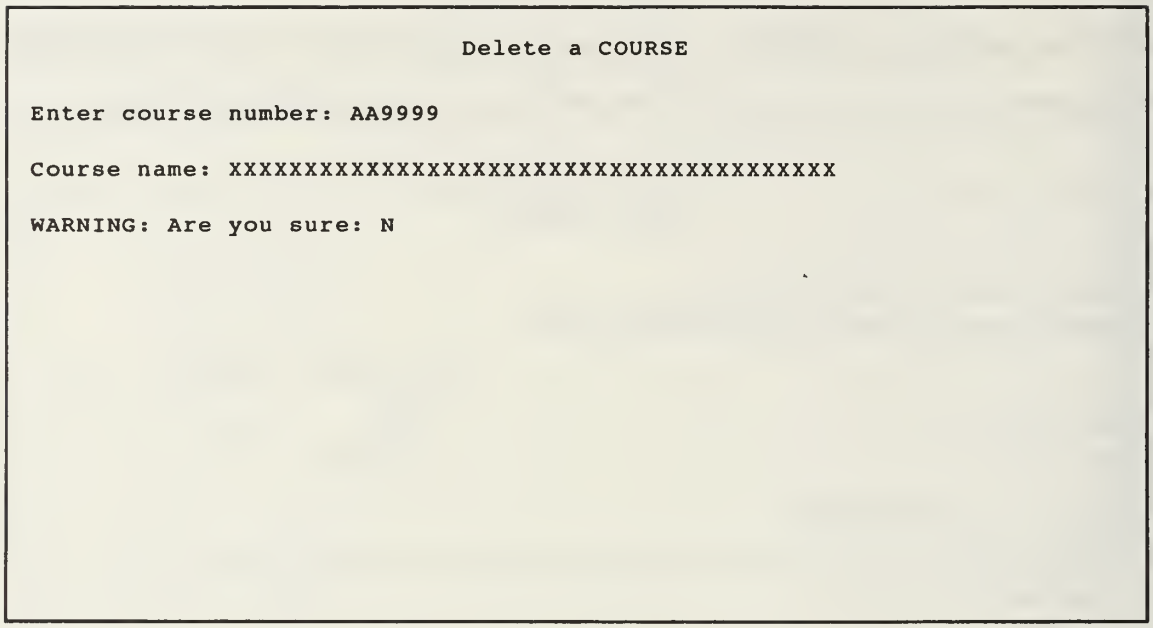
```
                              Add a COURSE


Enter course number: AA9999
Course name: XXXXXXXXXXXXXXXXXXXXXXXXXX

                                          Curriculum  Required   Qtr
                                            367          Y        3
Max size: 99                                620          N
Min size:  9                                817USMC      Y        1
                                            ...
When is course offered:
  Fall   : Y
  Winter: N
  Spring: N
  Summer: Y

Lecture hours: 9
Lab hours    : 9

Discipline: 999
```

Modify

This screen is identical in format to the *Add Course* screen. Enter the course number. The system will check to ensure that the number exists. After the course number is validated, you may view or edit any of the fields except course number.

Delete

Enter the course number. The system will check to ensure the course exists, and displays the course name. A message will appear asking you if you are sure. Deleting a course also deletes the fact that it is required for certain curricula. The *Delete Course* screen is displayed below.



List

You have the option of displaying a list of all courses to the screen, to the printer, or sending it a text file on disk. Printing the list prevents further use of your computer until printing concludes (unless you have a buffer or print spooler).

MAINTAIN CURRICULA

Curricular information changes infrequently. However, you still have the capability to manipulate curricular data.

Add

Enter the curriculum number. The system will check to ensure that the number does not exist. After the curriculum number is validated, enter the curriculum name, length, and the quarters in which the students start. The Add Curriculum screen is displayed below.

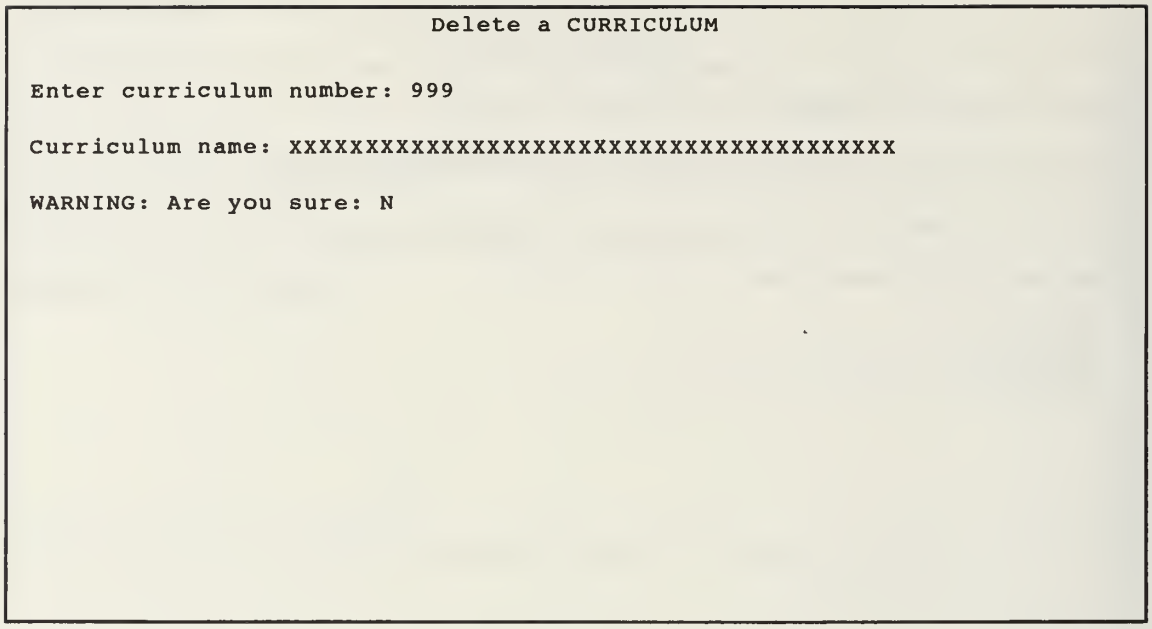
```
                        Add a CURRICULUM

Enter curriculum number: 999

Curriculum name: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Curriculum length : 9

Does curriculum start in:
  Fall   : Y
  Winter: N
  Spring: N
  Summer: Y
```

Modify

This screen is identical in format to the *Add Curriculum* screen. Enter the curriculum number. The system will check to ensure that the number exists. After the curriculum number is validated, you may view or edit any field except the curriculum number.

Delete

Enter the curriculum number. The system will check to ensure the curriculum exists, and display the curriculum name. A message will appear asking you if you are sure. CAUTION: Deleting a curriculum deletes its student sections and its associated course requirements. The *Delete Curriculum* screen is displayed below.



List

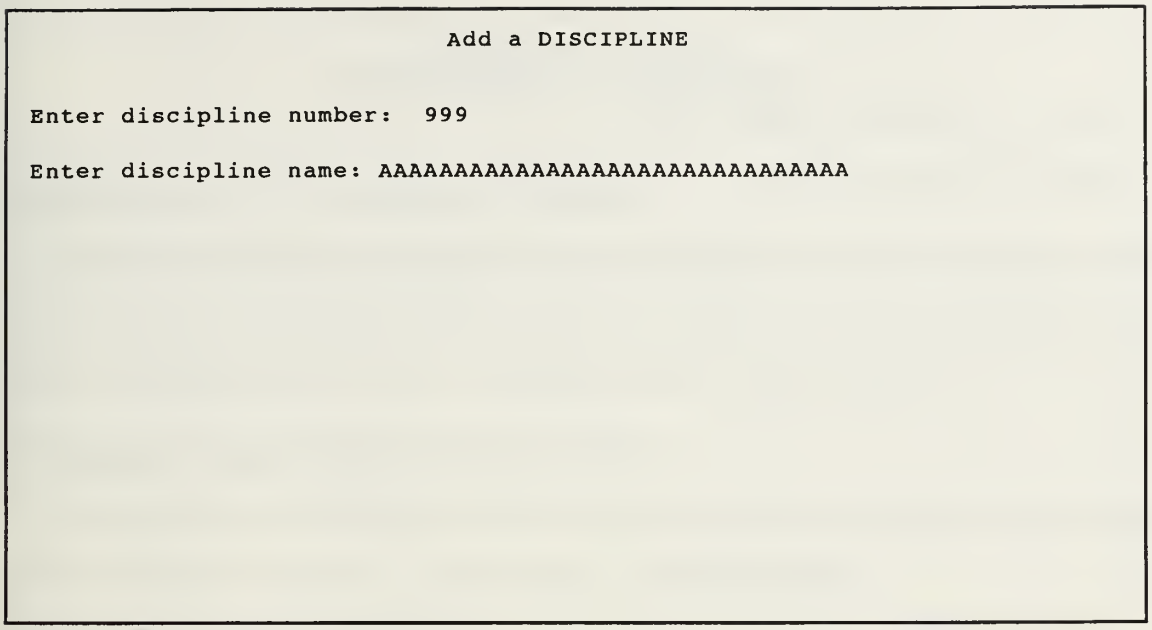
You have the option of displaying a list of all curricula to the screen, to the printer, or sending it a text file on disk. Printing the list prevents further use of your computer until printing concludes (unless you have a buffer or print spooler).

MAINTAIN DISCIPLINES

Discipline information changes infrequently. However, you still have the capability to manipulate this data.

Add

Enter the discipline number. The system will check to ensure that the number does not exist. After the discipline number is validated, enter the discipline name. The *Add Discipline* screen appears below.



Modify

This screen is identical in format to the *Add Discipline* screen. Enter the discipline number. The system will check to ensure that the number exists. After the discipline number is validated, you may view or edit the discipline name.

Delete

Enter the discipline number. The system will check to ensure the discipline exists, and displays the discipline name. A message will appear asking you if you are sure. The *Delete Discipline* screen is displayed below.

```
                    Delete a DISCIPLINE

Enter the discipline number: 999

WARNING: Are you sure: N
```

List

You have the option of displaying a list of all disciplines to the screen, to the printer, or sending it a text file on disk. Printing the list prevents further use of your computer until printing concludes (unless you have a buffer or print spooler).

PREPARING A SCHEDULE

GENERATING NEW COURSE SCHEDULE

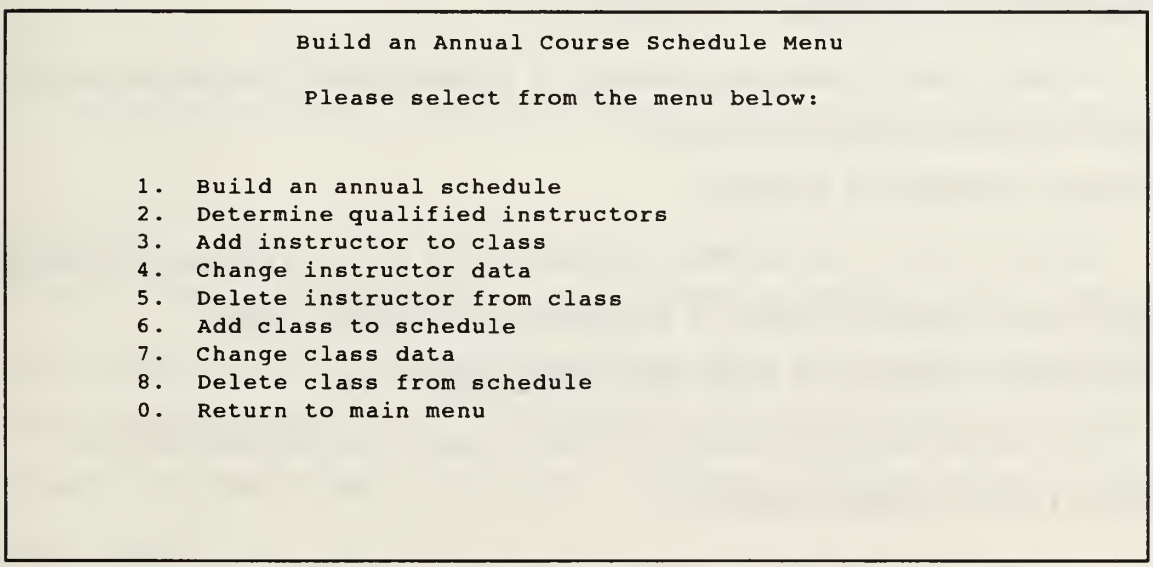
To generate a new course schedule, select menu option 3 at the main menu screen. The Build Annual Course Schedule menu will appear. Select 1 to build the schedule, and the system will automatically develop a teaching schedule (without instructors).

DETERMINE QUALIFIED INSTRUCTOR

Not implemented yet. This is an expert system that would automatically assign faculty based on historical instruction , areas of expertise, and availability.

MODIFYING THE SCHEDULE

This portion of the program will probably be used quite heavily. You may make changes to the schedule by selecting menu options 3-8, for the basic functions of adding, deleting, and changing instructor or class data.



REPORT GENERATION

The system offers a variety of reports. The first three are automated versions of existing reports. The remaining reports were developed to provide useful information to the decision maker. You can, of course, list any of the databases from within the database management application. Note: your computer may be limited to processing a report while it is printing. Background printing, spooling, and buffers may help.

ESTIMATED MAN-YEAR REPORT

This report provides an annual summary of the faculty teaching requirements (in man-years), broken down by discipline and quarter.

TEACHING SCHEDULE BY DISCIPLINE

This report lists all courses being taught for an entire year, sorted by discipline. Essentially, it is a QTS for four quarters.

QUARTERLY TEACHING SCHEDULE

This report lists all classes being taught in a particular quarter, and the names of the faculty members teaching those classes.

COURSE OFFERING REPORT

This report lists all courses offered and not offered by the Administrative Sciences Department in a particular quarter. It is disseminated throughout campus.

TEACHING SCHEDULE FOR ONE DISCIPLINE

This report is identical in format to the TSBD, except it lists only one discipline.

FACULTY TEACHING REPORT

This report provides a summary of all faculty personnel, and their teaching assignments throughout an academic year.

CURRENT

Selecting menu option 1 from the main menu displays the *Estimation* menu. A generated or copied schedule is required for a valid estimate. The Estimated Man-Year screen is displayed below:

```
                Estimated Man-Years for Academic Year 199X

Comments: Put your comments here

    Discipline:            Fall    Winter    Spring    Summer    Total
367 Information Systems    99.9    99.9      99.9      99.9      99.9
400 Management Policy      88.8    88.8      88.8      88.8      88.8
-

              Subtotal     99.9    99.9      99.9       99.9

              Grand Total  99.9
```

"WHAT-IF"

Select 2 from the Estimation menu. Enter the desired year and quarter. Curricula having sections that start in that quarter are displayed, one at a time. You can change the quota to provide the 'what-if' value. When competed with student estimates, you will be returned to the estimation menu. You can select 1 (EMY) or 5 (TSD) to review the results of your changes.

SAVING A SCENARIO

Select option 3 from the Estimation menu, and the program will prompt you to save the scenario (number between 1 and 9). When a what-if scenario is initiated, a unique scenario identifier (1-9) is established. Three of the files in the database are duplicated and renamed to match the scenario (e.g., SECTION2).

LOADING A SCENARIO

To load a previously-saved scenario, select menu option 4 from the Estimation menu. You will be prompted to enter the number of the scenario.

PRINTING

You can print a copy of the TSD using the "What-if" values by selecting option 5 from the *Estimation* menu.

REFERENCES

Date, C.J., *An Introduction to Database Systems*, Addison-Wesley Publishing Co., Inc., 1990.

Gilbert, Philip, *Software Design and Development*, Science Research Associates, Inc., 1983.

Kroenke, David M., and Dolan, Kathleen, *Database Processing*, MacMillan Publishing Co., 1988.

McLean, Ephraim R., and Sol, Henk G., editors, *Decision Support Systems: A decade in perspective*, North-Holland Publishing Co., 1986.

Powers, Michael J., Cheney, Paul H., and Crow, Galen , *Structured System Development*, Boyd & Fraser Publishing Co., 1990.

Senn, James, *Analysis and Design of Information Systems*, McGraw-Hill, Inc., 1984.

Simpson, Alan, *dBase IV 1.1 Programmer's Instant Reference*, Sybex, 1991.

Thierauf, Robert J., *Decision Support Systems for Effective Planning and Control: A case study approach*, Prentice-Hall, 1982.

Whitten, Jeffrey L., Bentley, Lonnie D., and Barlow, Victor M., *Systems Analysis & Design Methods*, Irwin, 1989.

BIBLIOGRAPHY

Andriole, Stephen J., editor, *Microcomputer Decision Support Systems: Design, Impementation, and Evaluation,* QED Information Sciences, Inc., 1986.

Carlson, E.D., and Metz, W., "Integrating Dialog Management and Database Management," *Information Processing '80,* North-Holland Publishing Co., 1980 .

Carrabis, Joseph-David., *dBase III Plus: Programmer's Library,* Howard W. Sams & Company, 1987.

Cerull, M.J., "Information Systems Success Factors," *Journal of Systems Management,* v. 31, no. 12, 1980.

Date, C.J., *An Introduction to Database Systems,* Addison-Wesley Publishing Co., Inc., 1990.

Donovan, J.J., "Database System Approach to Management Decision Support," *ACM Transactions on Database Systems,* v. I, no. 4, pp. 344-369, December 1976.

Elam, J., and Henderson, J., *Knowledge Engineering Concepts for Decision Support Systems Design and Implementation,* Western Periodicals, 1980.

Fenstermacher, Stephen M., *A Mechanized Decision Support System for Academic Scheduling,* Master's Thesis, Naval Postgraduate School, Monterey, CA, March, 1986.

Gilbert, Philip, *Software Design and Development,* Science Research Associates, Inc., 1983.

Ginzberg, Michael J., Reitman, Walter, and Stohr, Edward A., editors, *Decision Support Systems,* North-Holland, 1982.

Jones, Edward, *The dBase Language Reference,* Osborne McGraw-Hill, 1990.

Kroenke, David M., and Dolan, Kathleen, *Database Processing,* MacMillan Publishing Co., 1988.

Lee, Daniel T., "Database Design for Decision Support Systems," *Management and Office Information Systems,* Plenum Publishing Corp, v. I, 1983.

Lewandowski, A., and Stanchov, I., editors, *Methodology and Software for Interactive Decision Support,* Springer-Verlag, 1989.

McLean, Ephraim R., and Sol, Henk G., editors, *Decision Support Systems: A decade in perspective*, North-Holland Publishing Co., 1986.

Page-Jones, Meilir, *The Practical Guide to Structured Systems Design*, Yourdon Press, 1988.

Pendegraft, N., Parks, W., and Accola, W., "A Microcomputer based Inventory DSS," *Proceedings, 14th Annual American Institute for Decision Systems Meeting*, Western Periodicals, pp.34-36, 1985.

Powers, Michael J., Cheney, Paul H., and Crow, Galen , *Structured System Development*, Boyd & Fraser Publishing Co., 1990.

Prague, Cary N., and Hammitt, James E., *Advanced Programming with dBase III Plus*, Tab Books, Inc., 1987.

Prague, Cary W., and Hammitt, James E., *dBase IV 1.1 Programming*, Windcrest, 1991.

Rodeck, Renee L., *Design of a Generic DSS for use with FAMIS Sytem*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March, 1986.

Senn, James, *Analysis and Design of Information Systems*, McGraw-Hill, Inc., 1984.

Simpson, Alan, *dBase IV 1.1 Programmer's Instant Reference*, Sybex, 1991.

Simpson, Alan, *Understanding dBase IV*, Sybex, 1989.

Sprague, Ralph H. Jr., and Carlson, Eric D. , *Building Effective Decision Support Systems.*, Prentice-Hall, 1982.

Thierauf, Robert J., *Decision Support Systems for Effective Planning and Control: A case study approach*, Prentice-Hall, 1982.

Turban, Efraim, and Sepehri, Mehran, "Decision Support and Expert Systems in Manufacturing Management," *Proceedings, 14th Annual American Institute for Decision Systems Meeting*, Western Periodicals, pp. 402-404, 1985.

U.S. Department of the Navy, *Naval Postgraduate School Catalog*, Monterey, CA, 1990.

Whitten, Jeffrey L., Bentley, Lonnie D., and Barlow, Victor M., *Systems Analysis & Design Methods*, Irwin, 1989.

Yourdon, Edward, *Modern Structured Analysis*, Yourdon Press, 1989.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Professor Shu Sheng Liao, Code AS/Lc Department of Administrative Sciences Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor Daniel R. Dolk, Code AS/Dk Department of Administrative Sciences Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Captain Mark D. VanUs P.O. Box 519 Scott AFB, IL 62225	1
6.	Captain William G. Reagle 3116 May Street Castle Shannon, PA 15234	1

Thesis
V3215 VanUs
c.1 Design and implementation of a decision support system to aid in the forecasting and scheduling of Administrative Sciences courses.